[illegible]

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

04735

DETERMINATION OF TAFEL CONSTANTS
IN NONLINEAR POLARIZATION CURVES

by

Thomas Edward O'Loughlin

December 1987

Thesis Advisor: Jeff Perkins

Approved for public release; distribution is unlimited

T239114

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (*If applicable*) 69	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, California 93943-5000	7b. ADDRESS (*City, State, and ZIP Code*) Monterey, California 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11. TITLE (*Include Security Classification*)

DETERMINATION OF TAFEL CONSTANTS IN NONLINEAR POLARIZATION CURVES

12. PERSONAL AUTHOR(S)

O'Loughlin, Thomas E.

13a. TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (*Year, Month, Day*)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1987 December	140

16. SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	Tafel Constants, Corrosion, Potentiodynamic, FORTRAN Graphics

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

The presence of non-linear behavior in potentiodynamic polarization plots has resulted in difficulty in determining the Tafel constants from such plots. A FORTRAN based program involving numerical differentiation techniques was used to determine the existence of the Tafel regions.

Various alloys polarized in synthetic seawater and a 3.5 % NaCl solution were analyzed. Although severe concentration polarization often dominated the cathodic branches the techniques employed did allow for the selection of regions which approached linear behavior. The effects of concentration polarization in hindering the determination of Tafel constants were exemplified by the uncovering of a cathodic branch containing a small region where only activation polarization dominated followed by the onset and total domination of concentration polarization.

A method of determining where the anodic and cathodic currents begin to dominate the potentiodynamic polarization curve is introduced.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Jeff Perkins	22b TELEPHONE (*Include Area Code*) 408-646-2216 22c OFFICE SYMBOL 69Ps

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-243

Approved for public release; distribution is unlimited.

Determination of Tafel Constants in Nonlinear Polarization Curves

by

Thomas Edward O'Loughlin
Lieutenant, United States Navy
B.B.A., University of Notre Dame, 1981

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

ABSTRACT

The presence of non-linear behavior in potentiodynamic polarization plots has resulted in difficulty in determining the Tafel constants from such plots. A FORTRAN based program involving numerical differentiation techniques using a graphical display was used to determine the existence of the Tafel regions.

Various alloys polarized in synthetic seawater and a 3.5% NaCl solution were analyzed. Although severe concentration polarization often dominated the cathodic branches the techniques employed did allow for the selection of regions which approached linear behavior. The effects of concentration polarization in hindering the determination of Tafel constants were exemplified by the uncovering of a cathodic branch containing a small region where only actvivation polarization dominated followed by the onset and total domination of concentration polarization.

A method of determining where the anodic and cathodic currents begin to dominate the potentiodynamic polarization curve is introduced.

Thesis
C4735

TABLE OF CONTENTS

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5101

LIST OF SYMBOLS USED

cm^2	centimeter squared
C.R.	corrosion rate
E_{corr}	corrosion potential of a single metal
ϕ_{corr}	corrosion potential of a single metal
E	oxidation potential
ϕ	reduction potential
E.W.	equivalent weight
i_{corr}	corrosion current density of a single metal
i_a	current density
i_o	equilibrium exchange current density
i_L	limiting current density
mpy	mils per year
mV	millivolt
uA	microampere
β	Tafel slope
β_a	anodic Tafel slope
β_c	cathodic Tafel slope
η	polarization
η_a	activation polarization
η_c	concentration polarization
PAR	Princeton Applied Research
PDP	Potentiodynamic polarization

ACKNOWLEDGMENT

Special thanks to my wife Luanne whose patience was greater than my perseverance. To Professor Jeff Perkins and Mr. Tom Kellogg whose guidance and expertise was most appreciated.

I. INTRODUCTION

Recent studies at the Naval Postgraduate School have centered on the vibrational characteristics of certain high-damping alloys, [Refs. 1 - 8]. Most of these studies have been concerned with either the measurement of the damping capacities of these alloys and/or with the heat treatments necessary to attain the microstructures associated with the observed high-damping capacities. In addition, two separate studies were conducted in an attempt to determine the corrosion characteristics of these high damping alloys in marine environments.

In the studies involving corrosion characteristics, both laboratory and actual marine environment exposure experiments were conducted. The actual marine environment exposure tests calculated corrosion rates based on direct weight loss methods. The laboratory tests conducted were Linear Polarization and Potentiodynamic Polarization in synthetic seawater [Ref. 9] and a 3.5% NaCl solution [Ref. 10] environments. Although the correlation between the laboratory and direct exposure methods was good, the individuals involved in both of these laboratory studies expressed concern with the difficulty in determining the Tafel constants necessary to determine corrosion rates.

The actual experiments were conducted on the EG&G Princeton Applied Research Model 351 corrosion measurement system. The Model 351 utilizes a PAR Model 1000 microcomputer featuring two Motorola 68000 microprocessors and a touch-screen input to create an environment which is well suited for conducting various corrosion tests. The system is, however, limited by the lack of a key board and more importantly by an operating system which is proprietary in nature. These two combine to make any analysis of experimental data, other than by the means provided with the operating software, impossible unless the data can be transfered to another computer.

II. DATA TRANSFER

The PAR 351 and its associated software has the ability both to plot the results of an experiment and to send just the data points to a printer via an RS 232 serial port. Although the software supports only Houston Instruments plotters, the printer option is configurable for numerous printers as long as the printer supports DC1 (XON, ASCII 17) and DC3 (XOFF, ASCII 19) protocol. Other RS 232 requirements such as baud rate, parity, word length, and stop bits are configurable as the user sees fit.

During the initial data transfers, in the present work, Hayes Microcomputer Inc. SMARTCOM II modem software was used to capture the data. This method was successful most of the time but since the two systems were not using modems but instead were directly connected via their serial ports some problems were experienced and editing of the data file after transfer was always necessary. The most serious detraction was the need to wade through various menus in order to accomplish the transfer.

In view of the both the problems experienced with using SMARTCOM II and the neccessity for the user to be familar with the way SMARTCOM II works, it was decided to write a

data transfer/capture program which would both speed execution and eliminate the need for user knowledge of communications software.

Using the sample TTY data transfer program included in the manual for Microsoft GW-BASIC version 2 [Ref. 11] as a core a BASIC program was written to accomplish the transfer. The core was modified to eliminate the transfer portion and most importantly line feeds and carriage returns sent by the 351 were automatically deleted and a carriage return was inserted only at the end of a line containing data. This eliminated the problem of blank lines appearing within the data file corresponding to page headers and footers. Other modifications which were made involved checking for and correcting two formatting variations which resulted in compatibility problems between the transfered file and its use in FORTRAN application programs. The first involved a change from a floating format to a exponential format when the potential was equal to 0.0000 volts. The 351 normally sends potential with 4 significant digits. When the potential is 0.0000 volts it is displayed as -XXE-12. The formatting shift is not compatible with any FORTRAN formats as the total field width changes in addition to the minimum number of available digits. This occurance is now trapped and the file modified to reflect the actual potential of 0.0000 volts. The second formatting variation always occured and did so in a region which was of extreme interest

as follows. When the current density changes from negative to positive there is a range where the exponent is E-12. Unlike FORTRAN write statements which always reserve a space for the + or - sign, ie. the + sign is not printed unless specified, the 351 used the space for the - sign if present or for the first digit if not. Again this resulted in a change in total field width at the occurance of -a.bcdE-12. The FORTRAN read statment would read this to be -a.bcdE-1. This was handled by simply deleting the least significant digit such that the number is read as -a.bcE-12 when this occurs.

In its final form the data transfer program requires the user to reply only a to a (R)eceive or (E)xit prompt and to enter a file name if the (R)eceive option is selected. Because the 351 can not write a selected configuration to disk the user is still required to properly configure the 351 to the proper settings when powering up. These are:

```
PROTOCOL     - PRINT
PARITY       - NONE
BAUD RATE    - 2400
STOP BIT     - 1
WORD LENGTH - 8
```

Once these options are properly set on the 351 the desired data file should be loaded into the 351 and formatted into the linear option. At this point the user

selects the (R)eceive option on the Z248, presses the "PRINT DATA" prompt on the screen of the 351 and enters a filename on the Z248.

The final version of the data transfer program was compiled using Microsoft's BASIC compiler Version 1.0. By eliminating the writting of the data to the screen in addition to the file and using the compiled program vice the BASIC interpreter a file which will use 60K of disk space can now be transferred in under 5 minutes as opposed to the 20 minutes it took with SMARTCOM II. A listing of the program is included in Appendix B.

III. CORROSION RATES

A. THEORETICAL EQUATIONS

In general the corrosion behavior of a metal can be described by the use of the Nernst equation, activation potential, concentration potential and mixed potential theory.

The Nernst equation is a modification of the Gibb's Free Energy equation and expresses the tendency of a corrosion reaction to proceed in terms of EMF based on the activities of the products and reactants of the 1/2 cell.

$$E = E^{\circ} - \frac{RT}{nF}\log_{e}(Q)$$

E = Electrode oxidiation potential (ø for reduction)
E° = Equilibrium oxidiation potential (ø for reduction)
R = Gas constant
T = Temperature °K
n = Number of electrons taking part
F = The Farday
Q = Activities of products/activities of reactants

The potential of the cell is the algebraic sum of oxidation and reduction 1/2 cells.

Activation polarization describes the current density which exists when the cell is polarized.

$$i_{anodic} = i_o \exp(+ZF\alpha E/RT)$$

$$i_{cathodic} = i_o \exp(-ZF(1-\alpha)E/RT)$$

i_o = equilibrium exchange current density
Z = Valance of the active species
E = Polariztion about the equilibrium potential
α = fraction of E which applies to i_{anodic}
R,T,F as above

By manipulating the above equation the activation polarization η_a can be written as:

$$\eta_a = \beta \log_{10} (i_a/i_o)$$

$$\beta = 2.3RT/\alpha ZF$$

Although activation polarization occurs for both the anodic and cathodic reactions the cathodic reaction can also be effected by concentration polarization. When this occurs the reduction reaction occurs at such a rate as to decrease the concentration of ions in the solution near the surface of the metal. This results in a change in the potential of the 1/2 cell reaction since the effective activity in the Nernst equation changes.

The concentration polariztion $_c$ is expressed as:

$$\eta_c = \frac{2.3RT}{nF} \log_{10} (1 - i_a/i_L)$$

i_a = operating current density
i_L = limiting current density

The limiting current density is inversely proportional to the thickness of the diffusion layer. This thickness is dependent on various geometric and environmental factors and can often only be determined experimentally. The effect of concentration polarization can be minimized by agitating the solution, thus decreasing the layer thickness, or, as in the case of oxygen reduction, by heating the solution to reduce the concentration of the dissolved gas. Both of these means of dealing with concentration polarization may significantly alter the desired corrosion environment and distort the correlation between the experimental results and the actual in-service corrosion rates.

In the absence of concentration polarization a dynamic polarization diagram of a cell reaction would appear as in Figure 1. The slope of the anodic and cathodic curves $d\phi/d(\log_{10} i_a)$ would be given by β as described in the activation polarization expression.

Most cells are affected by concentration polarization such that the cathodic polarization $\eta = \Sigma\eta_i = \eta_a + \eta_c$. The effect of this is shown in Figure 2. It is obvious from

this that $\log_{10}(i_a)$ approaches $\log_{10}(i_L)$ the linear behavior of the polarization curve is distorted and in cases where i_L and i_o are not sufficiently separate, a linear region may not exist.

Mixed potential theory concerns concerns the immersion of a single piece of metal into an electrolyte and the subsequent reactions. The theory consists of two simple hypotheses.

1. Any electrochemical reaction can be divided into two or more partial oxidiation and reduction reactions.

2. There can be no net accumulation of electrical charge during an electrochemical reaction. [Ref. 12 : pp 314]

It is these hypotheses that allows for the determination of two important parameters necessary for estimating corrosion rates. When the cathodic branch of the 1/2 cell with the highest equilibrium potential intersects the anodic branch of the other 1/2 cell the corrosion potential ϕ_{corr} and the corrosion current density i_{corr} can be used to estimate corrosion rates.

When the corrosion system consists of only a single metal in contact with a solution which contains a single reduction/oxidation reaction the overall reaction is the linear sum of the reduction currents and the oxidation currents.

If the solution contains more than one reduction/oxidation reaction the overall reaction may proceed as the linear combination provided that the equilibrium exchange current densities of the secondary reduction reactions are within one order of magnitude of the main reduction reaction. The secondary oxidiation reactions proceed in the same manner. As a result the secondary redox reactions affect either, both, or none of the branches of the overall reduction/oxidiation reaction. [Ref. 13: pp. 199]

If concentration polarization is not present both the anodic and cathodic polarization curves would consist of a of horizontial lines, representing the current added by the reaction at that potential, and slanted lines of some slope β representing the activation polarization of all the oxidation or reduction reactions which are influential at that potential. In the presence of concentration polarization the possibility of introducing numerous non-linearities into the cathodic branch is distinct. As the overall reduction reaction is the sum of all the influential reduction currents any region could consist of zones where the activation polarization of the ith reaction dominates the concentration polarization of the (i-1)th reaction. This can make the potentiodynamic curve extremely difficult to analyze.

B. DETERMINATION OF CORROSION RATES

As previously described the presence of numerous electrodes and/or metals in the corrosion system can significantly alter the shape of the polarization curve. Despite this if it was possible to design a current measurement system that was capable of measuring only the anodic and cathodic currents the determination of i_{corr} could still be simply made since $\log_{10} i_{corr}$ would be at the intersection of the anodic and cathodic branches of the polarization diagram.

In the real world polarization curves take on a 'Y' shape as a result measuring instrument limitations and the complexity of the corrosion environment. ϕ_{corr} can be measured directly since it is the equilibrium potential of the system. As the potential is scanned towards ϕ_{corr} the net current ($i_{anodic} - i_{cathodic}$) is measured. Only when the potential is displaced far enough from ϕ_{corr} does one of the two currents overwhelm the other and does typical Tafel behavior take place.

The presence of numerous 1/2 cell reactions, concentration polarization and the need for one of the currents to overwhelm the other can all combine to make the determination of i_{corr} difficult.

If concentration polarization is not present at some $\delta\phi$ from ϕ_{corr} the polarization diagram should adopt true linear behavior. The amount of $\delta\phi$ at which the curve should adopt

linear behavior is in itself open to disscussion. Ailor states that $\delta\phi$ should be at least 50mV from ϕ_{corr} [Ref. 14: pp. 199]. Once the linear regions are found two methods can be employed to determine i_{corr}. The simplest is to draw two straight lines from the linear regions towards ϕ_{corr}. The two lines should intersect at i_{corr}. A second method is to measure the slopes β (Tafel constants) of the linear regions of the anodic and cathodic branches. These values can be used in conjunction with data determined from a polarization resistance test to determine the value of i_{corr} by use of the equation:

$$i_{corr} = \frac{\beta_c \beta_a}{2.3(\beta_a+\beta_c)PR}$$

If concentration polarization is predominant over the cathodic region of the polarization curve, the so-called 'knee' method is often employed. In this method a tangent is drawn on the knee of the cathodic branch such that the angles formed on both sides of the tangent with the cathodic branch are equal. When this has been accomplished either of the methods discussed in the previous paragraph may be used.

Many other methods have been developed to determine i_{corr} or to determine corrosion rates directly from polarization data. Princeton Applied Research uses a Chi-squared minimization technique. This technique is included in the software package which operates their Model 351 corrosion measurement system as PARCalc. By by their own

admission, since the program is terminated based on a change in the average value of the square of the relative deviations between two successive iterations, it is necessary to determine the Tafel region before starting the process. [Ref. 15: pp IX-10]

LeRoy [Ref. 16:pp 1006-1012] postulated the determination of Tafel Slopes based on polarization resistance techniques. Without repeating the entire discussion it is sufficient to say his method of determining Tafel slopes based on expected values from activation and concentration polarization expressions, based on the assumption that the parameters necessary to evaluate those expressions were available, did not accurately describe the observed values. [Ref. 17: pp 1988-1989]

It would also seem possible to solve the overvoltage expression for the terms in question. Recalling the overvoltage expression:

$$\phi - \phi_{corr} = \beta_c \log_{10}(i_a/i_o) + \frac{2.3RT}{nF}\log_{10}(1 - i_a/i_L)$$

The problem is that β_c, i_o and i_L all represent unknowns and the i_L term is located in such a manner as to complicate the solution.

The $\log_{10}(1 - i_a/i_L)$ term can be replaced by the series expansion:

$$\log_{10}(1+X) = 2.3\left[X - \frac{X^2}{2} + \frac{X^3}{3} - \frac{X^4}{4} \ldots + (-1)^{n+1}\frac{X^n}{n}\right]$$

where $X = -i_a/i_L$

The final equation to be solved would be:

$$\phi - \phi_{corr} = \beta_c \log_{10}(i_a) - \beta_c \log_{10}(i_o) + 2.3\left[\Sigma(-1)^{n+1}\frac{X^n}{n}\right]$$

The difficulty with this approach is that a sufficient number of terms must be carried in the series in order to accurately represent the $\log_{10}$ term. This causes two severe problems. The first is n+2 data points must be used to solve the system of equations. When one considers that i_a is normally on the order of $\mu A/cm^2$, exponentiating it to the nth degree can easily result in a matrix to be solved involving coefficients of 50 or 60 orders of magnitude. The end result is that while the system can be solved such that the coefficients determined will result in the accurate representation of the actual overvoltage, the coefficients themselves have been influenced by round-off error and catastropic cancellation, and represent merely numbers.

The other problem exists with attempting a solution of this form. As discussed previously with the possibility of mixed electrodes, the associated redox processes may or may not effect the overall behavior of the system. Even if the

limiting current densities i_L were all known, it would still be necessary to determine their effect on the overall process in order to include them in the system of equations to be solved. As a result, for m different electrodes, m experiments involving just the electrode and the metal would have to be carried out just to determine their influence. Consider the use of such a method with synthetic seawater. At least 10 different chemical substances could effect the overall reaction in addition to the major metal oxidiation and hydrogen reduction reactions. It should be obvious that if the limiting current densities are not known the size of the system of equations to be solved could easily exceed the number of available data points. More importantly, one would have to assess the advantages of performing a polarization in a mixed medium if the same test has to be performed separately on the individual components first.

C. CALCULATION OF TAFEL CONSTANTS

In order to determine the location, if one existed, of the anodic and cathodic Tafel regions two numerical differentation techniques were employed. These two methods were the Four Point Central Difference method and the use of a Cubic Spline interpolating polynomial. A graphical display of the derivatives allowed the user to determine the

region of linearity. Once this region was determined a first order linear regression is performed on the original data within the region to determine the Tafel slopes.

1. Four Point Central Difference Method

When the data points are equally spaced the Central difference method may be used to approximate the derivatives at a point. The use of this method can result in considerable time savings over the cubic spline method as the derivative can be calculated without having to resort to the solving of a system of equations for each point in question. The method is not as accurate as the cubic spline method and graphical representations of the derivatives will often contain more "noise" than those derived from the cubic spline method.

The four point Central Difference formula for computing derivatives is:

$$\frac{d\phi}{d(\log_{10} i_d)} = \frac{12h}{8(f_1 - f_{-1}) + (f_{-2} - f_2)}$$

Where:

$h = \phi_1 - \phi_{I+1}$

$f_n = \log_{10}(i_d)$ at the Ith location from $\log_{10}(i_d)_I$

The Central Difference method has an error term associated with it. In this case the error term is on the order of $1/h^4$. Although this may appear to be a substantial term, it should be remembered it is a possible error, the

method is only being used to determine the Tafel region and when used the Tafel constant is inverted so the error term would actually be h^4.

2. Cubic Spline Interpolating Polynomial

When trying to numerically determine derivatives of experimental data which may or may not be equally spaced the use of interpolating polynomials is often suggusted. One of the most accurate methods is to pass a cubic spline thru the data points and then to differentiate the resulting equation to arrive at the derivative at a desired point. [Ref. 18: pp. 242]

The advantage of using a cubic spline is that in deriving the system of equations to arrive at the constants $a, b, c,$ and d which form the equation:

$$y = a_I(x - x_I)^3 + b_I(x - x_I)^2 + c_I(x - x_I) + d_I$$

the system is formed in such a manner that the value of the function, first derivative and second derivative are the same for the pair of cubics which join at each point.

This requires that:

$$y' = 3a_I(x - x_I)^2 + 2b_I(x - x_I) + c_I$$

and $$y'' = 6a_I(x - x_I) + 2b_I$$

In order to simplify the the mathematics involved the equations for the system are written in terms of the second derivatives of the interpolating cubic S_I.

The resulting interpolating cubic will then take on the form of:

$$h_{I-1}S_{I-1}+2S_I(h_{I-1}+S_I)+h_IS_{I+1}=6\frac{(y_{I+1}-y_I)}{h_I}-\frac{(y_I-y_{I-1})}{h_{I-1}}$$

where $h_I = x - x_I$

Solving the system leads to the following operations to determine the values of a_I, b_I, c_I and d_I.

$$a_I = (S_{I+1} - S_I)/6h_I$$
$$b_I = S_I/2$$
$$c_I = ((y_{I+1} - y_I)/h_I) - ((2h_IS_I + h_IS_{I+1})/6)$$
$$d_I = y_I$$

The resulting system of n - 2 equations in S_I involves n pairs of data points in order to generate the required number of equations. To arrive at the two additional equations to solve for S_1 and S_n, constraints are specified which pertain to the conditions at the ends of the curves. The three choices for the end conditions are:

1. $S_1 = S_n = 0$, which implies that the end cubics approach linearity at their extremities.

2. $S_1 = S_2$, $S_n = S_{n-1}$. This assumes that the end cubics approach parabolas at their extremities.

3. S_1 is a linear extrapolation of S_1 and S_2. S_n is a linear extrapolation of S_{n-2} and S_{n-1}.

In this case since the interest is in determining the derivatives $d\phi/d(\log_{10} i_d)$ pointwise, and not in developing an interpolating polynomial, it was decided to use the end conditions of $S_1 = S_n = 0$, and to use enough data points as to force the inconsistences caused by the assumed end conditions away from the point in question. Four data points on each side of the point in question were used in determining the the derivative at the point in question. This resulted in the solving of a 7 x 7 system of equations for each data point. By locating the point in question in the center of the data set, the accuracy of the estimate involved was improved, and the mathematics involved in determining the derivative were simply solving for c_1.

4. Linear Regression

Once the anodic and cathodic Tafel regions have been determined, the data points which comprise these regions are used in a least squares curve fitting routine to determine an equation of the form:

$$\phi = \beta(\log_{10} (i_d)) + b,$$

where β represents the Tafel slope. The values of a and b are found by solving the two following equations simultaneously.

$$\beta\Sigma(\log_{10}(i_d)_I)^2 + b\Sigma(\log_{10}(i_d)_I) = \Sigma(\log_{10}(i_d)_I)(\phi_I)$$

$$\beta\Sigma(\log_{10}(i_d)_I) + bN = \Sigma(\phi_I)$$

where N is the number of points in the Tafel region.

5. Considerations Affecting Method Selection

When the PAR 351 Corrosion Measurement system scans the potential during a Potentiodynamic test the current is measured at intervals of 2mV or 0.5mV of potential, depending on the relative change in the current. The sampling rate is normally 0.5mV when the current shows a substantial change and 2mV when it does not. This usually results in a sampling rate of 0.5mV in the Tafel regions, although severe concentration polarization in the vicinity of i_{corr} may result in a limited number of points having 0.5mV intervals on the cathodic branch. Two other aspects may also effect the potential intervals at which the current appears to have been sampled.

When data is sent out the serial port, the 351 normally uses a format similar to the FORTRAN Ga.b type. Not counting leading zeros, four significant figures are used. This results in a problem in the recorded potential values. When the potential is greater than or equal to 1.0 volts, or less than or equal to -1.0 volts, only three decimal places are carried. As a result, when the values of ϕ_{corr} fall in this range and the scan rate is 0.5mV, the recorded data seems to reflect that the potential has not changed for two successive current measurements.

Since the potential is being scanned it is rather obvious when this occurs that the recorded data does not accurately represent the experimental data. Since this

pattern is easily recognized, the array containing the data can be manipulated to more accurately reflect the experimental data.

Occasionally the potentials recorded display no regular pattern and the data, although questionable in nature, must be used as recorded.

When calculating the derivatives, the anodic and cathodic branches are performed separately. The calculations begin at the third data point away from E_{corr} when the Central Difference method is used, and at the fifth when using the cubic spline method. The calculations continue as long as the data points continue to suggest a slope of of the expected sign (+/-). This automatically excludes the passivating region of the anodic section, as it should since it is not part of the Tafel region, and excludes the possibility of the program terminating execution if it were to calculate an infinite slope. In general this strategy worked as expected and provided enough data points to properly examine the Tafel regions. The only experiment in which it failed was for the SONOSTON sample tested in 3.5% NaCl solution. As can be seen by examining the Potentiodynamic curve of this sample (Figure 3), the cathodic branch exhibited behavior throughout which caused either method to terminate execution without generating any derivatives. The reason for this behavior is unknown.

IV. GENERAL OVERVIEW OF THE PROGRAM

The program was developed so as to allow for its use without the supposition that the user was familar with the FORTRAN langauge or with personal computers in general. Numerous efforts were made to minimze the possibilty of the program prematurely terminating execution in the event of improper input or through floating-point operations resulting in invalid expressions. By taking advantage of the color capability of the monitor screen, colors are changed with respect to the type of input expected, and to display messages resulting from either input errors, data file format errors or a failure to properly follow the steps necessary for proper program execution.

The program was written using Microsoft FORTRAN Optimizing Compiler version 4.0. Graphics were achieved using Microcompatibles, Inc.'s GRAFMATIC library. Plotter support was achieved using Microcompatibles, Inc.'s PLOTMATIC library.

A summary of the main program and its subroutines is given below. The main program and its called subroutines are listed in Appendix C. The subroutines are presented in the order in which they appear in the main program.

A. START.FOR

This is the main program. The name of the first data file to be used is entered. Following calls to DATAIN, CHECK and DATAM a menu consisting of 6 options is presented:

1. Plot the potentiodynamic curve
2. Generate Tafel Slopes using cubic spline
3. Generate Tafel slopes using central difference
4. Use another data file
5. Overlay two potiodynamic curves
6. Exit

Option 1 generates calls to GRAPH1 and PLOT

Option 2 calls SLOPE1. Following this a first order linear regression of the anodic and cathodic branches using locations within the original array determined during a call to GRAPH1 is performed and the results along with original potentiodynamic curve are passed to GRAPH1.

Option 3 calls SLOPE. Upon return from SLOPE this option proceeds as Option 2

Option 4 returns the user to the 3rd executable line of START.

Option 5 calls DATAIN, CHECK and DATAM while retaining the data from the first file.

Option 6 obviously exits the program.

B. DATAIN.FOR

This opens the input file and, using a formatted read statement, generates the array containing potential and current density. Additionally this subroutine counts the total number of lines in the file. This is done to ensure

that formatting changes within the data file which were not corrected during the data file transfer will not have not caused the array generation to prematurely terminate. If a formatting error has been detected, an error message is displayed in red and the line number within the file is identified. This allows the user to exit the program and correct the error before continuing.

C. DATAM.FOR

This converts the absolute value of the current density to its $\log_{10}$ value and determines the position of E_{corr} with the array.

D. CHECK.FOR

This subroutine determines the number of data points in the anodic and cathodic branches to be used in the determining the derivatives when using the central difference method. This is done by checking log current density for values which might result in the calculation of an infinite slope (normally the onset of pitting or passivation) and for equal increments in the recorded potentials. Occasionally the recorded potentials will not reflect the polarization which has taken place. This occurs when the absolute value of the recorded potential is equal to or greater than 1.0. In these ranges the 0.5mV voltage

increment falls outside of the total width of the field and is not recorded. As this is a recognizable pattern, at the users option the problem can be corrected.

E. SLOPE.FOR

This is the first subroutine called when using the Central Difference method. It assigns graph titles, axes labels and calls DATADEL.FOR.

F. DATADEL.FOR

Using the points determined in CHECK, this calculates the numerical derivatives using the Central Difference method.

G. SLOPE1.FOR

This assigns graph titles and labels. It also determines the number of points, MI, to be used on both branches in the cubic spline method by checking only for values of $\log_{10}$ current density which could result in the calculation of slopes of infinite value. It calls CSPLIN. separately for the anodic and cathodic branches. MI anodic is not necessarily equal to MI cathodic.

H. CSPLIN.FOR

For MI number of points, this generates the 7x7 matrix and the 7x1 vector of values described in the cubic spline section MI-9 times. As each matrix is formed it is solved

by a call to LINSY1.FOR. Upon return from LINSY1 the 7x1 vector contains the solution which is used to calculate the derivative.

I. LINSY1.FOR

Using partial pivoting this subroutine solves the augmented 7x8 array by Gaussian Elimination. The original 7x1 vector contains the solution [Ref. 19].

J. GRAPH1.FOR

This is the graphics output of the selected main menu option. Through the use of the numeric keypad as softkeys the user can scale the axes as desired. The option to return to the original graph or the previous display always exists. When plotting the results of either numeric differentation method, only those points which consecutively fall within the desired region are plotted. Once the axes have been scaled such that graph contains only the linear region of the anodic or cathodic branches the position of the start and stop points within the original array containing the derivatives are saved. Since, when calculating the derivatives, the data point containing Ecorr is always used as a reference, ie. (depending on the method) the first derivative for each initial branch is either 3 or 5 points away from E_{corr}, the start and stop points also represent the position of the Tafel regions within the original potential and log10 current density array. When

the graph is not of the numerical derivatives, any data point which falls within the desired axes is plotted. If desired GRAPH1 calls PLOT.FOR for a hard copy output of what is currently displayed on the screen.

K. PLOT.FOR

This plots the graph displayed on the screen. Although many of the arguments passed from GRAPH1 are used directly as passed, this is not a 'screen dump'. The resolution of the plotter is much higher than that of the screen. As a result what appears to be a one pixel jump on the screen will not plot as such. Two sets of tabulated data may be plotted on each set of axes. At the users discretion different color pens may be used to plot the axes, first output, second output, and labels.

L. CONFIGURATION

In its final form, the program has been compilied and linked into an executable file which runs on a Zenith 248 computer which is equipped with a 80286 microprocessor, an 80287 Numeric Processor Extension, a EGA graphics card with 64K video RAM, and a color monitor which supports the EGA color mode. The plotter used is an Houston Instruments DMP-40 series digital plotter.

The plotter library is plotter series specific. As a result of this, the use of a plotter other than a Houston Instrument's DMP-29 or greater would require obtaining

another library from Microcompatibles Inc. As many of the subroutines contained within PLOTMATICS are similar to those called within GRAFMATICS, it can probably be safely assumed that little if any editing of the subroutine PLOT.FOR would be required.

To use the program with a video configuration other than that previously described would require editing of the subroutines PLOT.FOR, GRAPH1.FOR, DATAIN.FOR, START.FOR and CHECK.FOR. DATAIN, START and CHECK would require editing only if the video configuration was monochrome. PLOT and GRAPH1 would require editing for any graphics confiquration other than EGA.

V. RESULTS AND DISCUSSION

A. SYNTHETIC SEAWATER

Table 1 below compares the Tafel constants determined by using the new program with those obtained by Escue [Ref. 9].

TABLE 1

TAFEL CONSTANTS OBTAINED FROM SYNTHETIC SEAWATER EXPOSURE

MATERIAL	β_a	β_a*	β_c	β_c*
Ti-50Ni	0.1400	0.174	0.1155	0.132
1020 Steel	0.0758	0.068	0.1481	0.114
304 Steel	0.0241	0.178	0.0207	
7075 Al	0.3790	0.087	0.1584	0.062
Fe-Cr-Mo	0.1775	0.121	0.1226	0.141
Fe-Cr-Al	0.1840	0.180	0.1369	0.151
Cu-Mn-Al-Fe-Ni	0.0399	0.048	0.1579	0.120
Cu-Mn-Al	0.0633	0.035	0.0652	0.045
Cu-Zn-Al	0.0804	0.062	0.0785	0.075
630 Bronze	0.0403	0.031	0.0724	0.041

* From Escue [Ref. 9]

Table 2 below compares the current densities and related corrosion rates of the samples tested in synthetic seawater. The first line for each material represents the current density as determined by the intersection of the Tafel lines. In many of the cases the anodic and cathodic Tafel lines did not intersect at a single point and are

joined by a horizontal line. In these cases i_{corr} was determined by taking the average of the anodic and cathodic i_{corr} at the point where the lines intersected a horizontal line form by ϕ_{corr}.

The second line represents the results obtained by combining the Tafel slopes with the polarization resistance data.

TABLE 2

CURRENT DENSITY AND CORROSION RATES

MATERIAL	i_{corr} uA/cm^2	i_{corr}* uA/cm^2	CR mpy	CR* mpy
Ti - 50%Ni	1.245	1.734	0.573	0.799
	4.182	4.960	1.822	2.160
1020 Steel	4.212	3.837	1.981	1.805
	20.325	17.201	9.580	8.110
7075 Al	1.413	0.421	0.591	0.176
	3.067	0.994	1.281	0.415
304 Steel	0.106	0.728	0.048	0.329
	0.157	1.190	0.071	0.537
Fe-Cr-Mo	0.750	0.804	0.354	0.380
	2.294	2.060	1.085	0.974
Fe-Cr-Al	0.822	0.830	0.362	0.365
	1.183	1.240	0.520	0.545
Cu-Mn-Al-Fe-Ni	8.241	8.854	4.165	4.070
	9.662	10.400	4.882	5.260
Cu-Mn-Al	2.089	1.119	1.111	0.595
	1.459	0.894	0.784	0.481c
Cu-Zn-Al	2.239	1.897	1.136	0.963
	3.780	3.230	1.919	1.640
630 Bronze	1.762	1.442	0.901	0.737
	6.878	4.690	3.550	2.420

* From Escue [Ref. 9]

1. Ti-50%Ni (TINILG)

These results compared favorably with those obtained by Escue. Figure 4 shows the derivatives as calculated by the cubic spline method. As can be seen in this figure no true linear regions are distinguishable in either of the two branches. When this occurs the "knee" method can be employed by examining the branches for a range where the derivatives oscillate about a linear midpoint. In this case both the anodic and cathodic branches showed this oscillation in the log current density range of -5.28 to -5.51. Figure 5 shows the resultant Tafel lines plotted on the original potentiodynamic curve. Although the curves intersect ϕ_{corr} at what could be considered to be a single point the i_{corr} determined in this manner differs greatly from that calculated by using the Tafel constants and the polarization resistance data.

2. 1020 Carbon Steel (S1020LG)

Figures 6 and 7 show the results as obtained for the 1020 carbon steel sample. Of all of the samples tested this one exhibited a cathodic branch with the least tendency to behave in a linear fashion. If there was any linear behavior in the cathodic branch it occured at a $\log_{10} i_{corr}$ of about -5.2. The anodic branch displayed a much longer lasting region around $\log_{10} i_{corr} = -4.3$. This sample also showed the greatest disparity in the value of i_{corr} depending on the manner in which it was calculated.

3. 7075 Aluminum (AL7075LG)

In this case the central difference method showed a linear region in both the anodic and cathodic branches. As can be seen in Figure 8 the linear regions exist at the pronounced peaks in the derivatives. In this case the Tafel lines did intersect at a single point as seen in Figure 9.

4. 304 Stainless Steel (SST304LG)

The calculated corrosion rates and the respective Tafel constants differed greatly from those obtained by Escue. Although the calculated corrosion rates correlated better to the actual weight loss methods, the need to terminate the calculating of derivatives before the onset of passivity may have caused a linear region in the anodic branch to be omitted. These points can be seen by the sharp peak on the anodic branch on Figure 10, and the corresponding point on the potentiodynamic curve Figure 11. The Tafel lines intersected at a single value of $\log_{10} i_{corr}$ but this time there was a good correlation between the calculated and graphical i_{corr}.

5. Fe-Cr-Mo (VACR01LG)

Again the presence of any truly linear regions is questionable, although both branches, Figure 12, show a range where the deviations from linear behavior are minimal. Figure 13 once again shows a theoretical intersection of the Tafel lines.

6. Fe-Cr-Al (VACRO2LG)

This alloy showed behavior similar to the Fe-Cr-Mo alloy described above. What may be considered a truly linear region in the cathodic branch is shown in Figure 14 just to the left of the sharp peak. The overlay of the Tafel lines on the potentiodynamic curve is shown in Figure 15. The Tafel constants derived show a closer relationship to those calculated by Escue, than did a comparision of the Fe-Cr-Mo sample.

7. Mn-Al-Fe-Ni (SONOSTLG)

SONOSTON, Figures 16 and 17, displayed a linear anodic branch and, as did most of the alloys, a cathodic branch effected by concentration polarization.

8. Cu-Mn-Al (INCRLG)

The result here again showed a linear anodic branch but in this case the concentration polarization did not overwhelm the cathodic branch in such a manner as to distort the region where activation polarization is predominant. Figure 18 shows the graph of all the derivatives and Figures 19 and 20 show the graphs of the derivatives in what was decided to be the Tafel region. Figure 21 shows the original potentiodynamic and the Tafel lines.

9. Cu-Zn-Al (DLALCLG)

Experinced similar behavior as the Cu-Mn-Al sample. Once again a linear region was very apparent in the anodic branch. Figures 22 and 23.

10. 630 Series Bronze (BRNZLG)

This was the only sample of those tested in the synthetic seawater where the Tafel regions for the anodic and cathodic branches were found in the same region of log i_{corr}. As seen in Figure 24 concentration polarization again quickly dominates the cathodic branch. Figure 25 displays the overlay.

B. 3.5% NaCl SOLUTION

In a similar fashion Table 3 lists the Tafel constants and Table 4 the resultant current densities obtained from the samples tested in the 3.5% NaCl solution. In general the current results show a stronger correlation with the corrosion rates obtained by Akthar [Ref. 10] than did the comparision of the current results with those obtained by Escue [Ref. 9] for the synthetic seawater.

TABLE 3

TAFEL CONSTANTS OBTAINED FROM
3.5 % NaCl SOLUTION EXPOSURE

MATERIAL	β_a	β_a*	β_c	β_c*
304 Steel	0.1259	0.4117	0.0951	0.117
7075 Al	0.0116	0.010	0.0383	2.36
Fe-Cr-Mo	0.2821	0.234	0.1053	0.110
Fe-Cr-Al	0.1203	0.214	0.1292	0.1285
Cu-Mn-Al	0.0082	0.0137	0.0337	1.379
630 Bronze	0.0524	0.570	0.1472	0.321

* From Akthar [Ref. 10]

TABLE 4

CURRENT DENSITY AND CORROSION RATES

MATERIAL	i_{corr} $\mu A/cm^2$	i_{corr}* $\mu A/cm^2$	CR mpy	CR* mpy
7075 Al	6.960	7.406	0.048	2.965
	10.460	11.700	1.842	4.685
304 Steel	0.052	0.100	0.026	0.050
	0.012	0.020	0.006	0.010
Fe-Cr-Mo	0.043	0.040	0.019	0.018
	0.609	0.613	0.274	0.276
Fe-Cr-Al	0.493	0.660	0.234	0.313
	0.916	1.180	0.428	0.551
Cu-Mn-Al	2.864	6.062	1.454	3.078
	5.251	10.800	2.666	5.483
630 Bronze	0.093	0.129	0.048	0.066
	3.570	4.480	1.842	2.312

*From Akthar [Ref. 10]

1. 7075 Aluminum (AL7075)

In this case the effect of concentration polarization made any estimate of the cathodic Tafel constant extremely difficult. As seen in Figures 26 and 27 concentration polarization completely overwhelmed the cathodic branch. As a result only the "knee" method cuold be used, but its accuarcy is questionable. The anodic branch displayed typical linear behavior.

2. 304 Stainless Steel (SST304)

The cathodic branch of this sample did display a linear region located between the sharp peaks in Figure 28. The anodic branch displayed numerous attempts to passivify as seen in Figure 29.

3. Fe-Cr-Mo (VCMOPD)

Similar behavior to the 304 Stainless steel above. The bandwidth exhibited by the cathodic branch near $\log_{10} i_{corr}$ of about - 6.5 Figure 30, typifies the knee in the curve. Again numerous attempts to passivify limited the range of data which could be use in the anodic branch Figure 31.

4. Fe-Cr-Al (VACAL2)

Both the anodic and cathodic regions, Figure 32, displayed a region where the behavior could be considered to be linear. Although the Fe-CR-Mo and Fe-Cr-Al samples tested in synthetic seawater displayed similar behavior, the Fe-Cr-Al sample tested in the 3.5% NaCl solution did not exhibit the tendency to passivify, Figure 33, as did the Fe-Cr-Mo sample.

5. Cu-Mn-Al (INCRMTE)

This sample best typifies the problem associated with concentration polarization. On Figure 34 a linear region can be seen in the cathodic branch at a $\log_{10} i_{corr}$ of about -5.3. This region is followed by a rapid decrease in the slope of the curve indicative of concentration polarization. When the original potentiodynamic curve is viewed thru out its entire polarization range, Figure 35, very little can be determined due to the great changes in current density on the anodic branch. When the range in the

cathodic branch displaying the linear bevaior is scaled properly a textbook example of concentration polarization is seen, Figure 36. The overlay is shown in Figure 37.

6. 630 Series Bronze (BRZ630PD)

As seen on Figures 38 and 39 this sample experienced similar behavior to the one above. All though the linear region directly before concentration polarization is not as well defined the cathodic behavior is dominated first by activation polarization and then by concentration polarization.

C. COMPARISION OF CURRENT DENSITIES

In many of the samples tested the corrosion rates as determined by the intersection of the Tafel lines were significantly lower than those obtained by the use of the Tafel Slopes and linear polarization data. As previously stated the Tafel regions should begin to dominant the polarization curve at an polarization increment of about 50mV from E_{corr}. This implies that a linear region should not be present at increments less than this. With these tests a linear region was often found within this zone. Because of the general shape of a potentiodynamic polarization curve the resultant lines drawn from this region will always have a lower slope and thus intersect at a lower value of i. The disparity underscores the need to use the average of i_{corr} as determined by the intersection

of the Tafel lines and that calculated by the Tafel slopes and linear polarization data when estimating corrosion rates.

Another interesting feature was noticed when conducting some data manipulation of the potentiodynamic polarizations analyzed. When the absolute value of ($\phi - \phi_{corr}$) was plotted against $\log_{10} i_a$ in many of the cases the anodic and cathodic branches began to deviate significantly at a log i_a corresponding to the intersection of the two Tafel lines. This may be coinicidental but also may warrant further discussion. When a linear polarization resistance test is conducted the potential is scanned with a range of ϕ_{corr} +/- 25 mV. The polarization resistance is determined by calculating the slope of the linear region about i = 0 A/cm2. If a linear polarization curve is plotted in the same manner as above, while neglecting the singularity at i = 0 A/cm^2, the deviation discussed would determine the end of the linear region. Returning to the equation used to determine i_{corr} using the Tafel constants and the polarization resistance it can be shown that in order to the polarization resistance to be a constant that $\beta_a\beta_c = C*(\beta_a+\beta_c)$, where C is a constant of proportionality. The current being measured during a potentiodynamic polarization is equal to the difference between the anodic and cathodic currents. Only when one of the currents begins to overwhelm the other would there be a substantial change in the

measured current with respect to a given overvoltage. This would correlate to the deviation between the anodic and cathodic branches on a plot of the absolute value of ($\phi - \phi_{corr}$) vs log (i_a). It also should correspond to the end of the linear region on the linear polarization curve since at this point the constant of proportionality would no longer be a constant as one of the two branches is beginning to overwhelm the other. It is postulated that a technique of this method could be used to determine the minumium overvoltage at which the search for a Tafel constant could begin. It is recognized that this does not account for the possibility of completely symetric anodic and cathodic branches but if this was the case there would be no problem in resorting to the +/- 50mV rule proposed by Ailor.

VI. CONCLUSION AND RECOMMENDATIONS

The FORTRAN based graphics program proved to be extremely valuable in determining the existence of linear Tafel regions particularly in those cases where concentration polarization dominated the cathodic branch.

With the ability to transfer the experimental data to a programable computer the opportunity exists to further analyze the data. This could allow for the determination of which electrodes are effecting the main redox reaction when the environment consists of a solution of mixed electrodes. In order to accomplish this it is recommended that a single high damping alloy be subject to individual potentiodynamic polarizations containing a single component, at its normal concentration, of synthetic seawater. Through the use of graphical techniques it should be possible to isolate the influential redox reactions.

In order to accomplish this it will be necessary to conduct all the experiments in solutions which contain the same amount, or at least a monitored amount, of dissolved oxygen. This is one factor which was not considered in the previous two experiments and may have made the analysis more difficult than necessary.

In lieu of this approach an effort should be made to determine the limiting current densities causing the severe concentration polarization.

LIST OF REFERENCES

1. Dew, D.D., Strain Dependent Damping Characteristics of a High Damping Mn-Cu Alloy, M. S. Thesis, Naval Postgraduate School, Monterey, California, September 1986.

2. Reskusich, J., Cyclic Strain Amplitude and Heat Treatment Effects on the High Damping Behavior of INCRAMUTE Alloy Under Random Vibrational Loading in the 50 -1000Hz Frequency Range, M. S. Thesis, Naval Postgraduate School, Monterey, California, September 1986.

3. Leary, L.W., Low Temperature Effects on High Damping in the INCRAMUTE and SONOSTON Cu-Mn Based Alloys, M. S. Thesis, Naval Postgraduate School, Monterey, California, December 1986.

4. O'Toole, J.F., Heat Treatment Effects on High Damping in VACROSIL, an Fe-Cr based Alloy, M. S. Thesis, Naval Postgraduate School, Monterey, California, December 1986.

5. Ferguson, D. B., Characterization of Damping in Fe-Cr Based and Other Quiet Metal Alloys, M. S. Thesis, Naval Postgraduate School, Monterey, California, March 1987.

6. Cronauer, J.T., A comparision of High Damping Shape Memory Alloys with Cu-Mn-Based and Fe-Cr-Based Alloys, M. S. Thesis, Naval Postgraduate School, Monterey California, June 1987.

7. Patch, G.R., Computerization of Data Acquisition, Data Analysis and Data Correlation for High Damping Alloys, M. S. Thesis, Naval Postgraduate School, Monterey. California, September 1987.

8. Childs, J.L., The Effect of Heat Treatment and Cyclic Strain Amplitude on the Damping Properties of Iron-Chromium Aloys, M. S. Thesis, Naval Postgraduate School, Monterey, California, December 1987.

9. Escue, W., Characterization of the Corrosion Behavior of Various High Damping Alloys, M. S.Thesis, Naval Postgraduate School, Monterey, California, June 1987.

10. Akthar, S., Corrosion Performance of High Damping Alloys in a 3.5% NaCl Solution, M. S. Thesis, Naval Postgraduate School, Monterey, California, September 1987.

11. Zenith Data Systems, Microsoft GW BASIC Interpreter Version 2,,1984.

12. Fontana, M.G. and Greene, N.D., Corrosion Engineering, McGraw-Hill, 1978.

13. Fontana, M.G. and Steele, R. W., Advances in Corrosion Science and Technology, Plenum, 1977.

14. Ailor, W.H., Handbook on Corrosion Testing and Evaluation, John Wiley and Sons, 1975.

15 EG&G Princeton Applied Research, Model 351 Corrosion Measurement System, 1985.

16. LeRoy, R.L., "Evaluation of Corrosion Rates from Nonlinear Polarization Data", Journal of the Electrochemical Society, v. 124, July 1977.

17. Palombarini, G., Cammarota, G.P., and Felloni, L., An untitled dicussion on reference 16 above, Journal of the Electrochemical Society, v. 125, December 1978.

18. Gerald, C.F. and Wheatley, P.O., Applied Numerical Analysis, Addison-Wesley, 1984.

19. Nyhoff, L. and Leestma, S., FORTRAN for Engineers and Scientists, pp. 263-265, Macmillan, 1985.

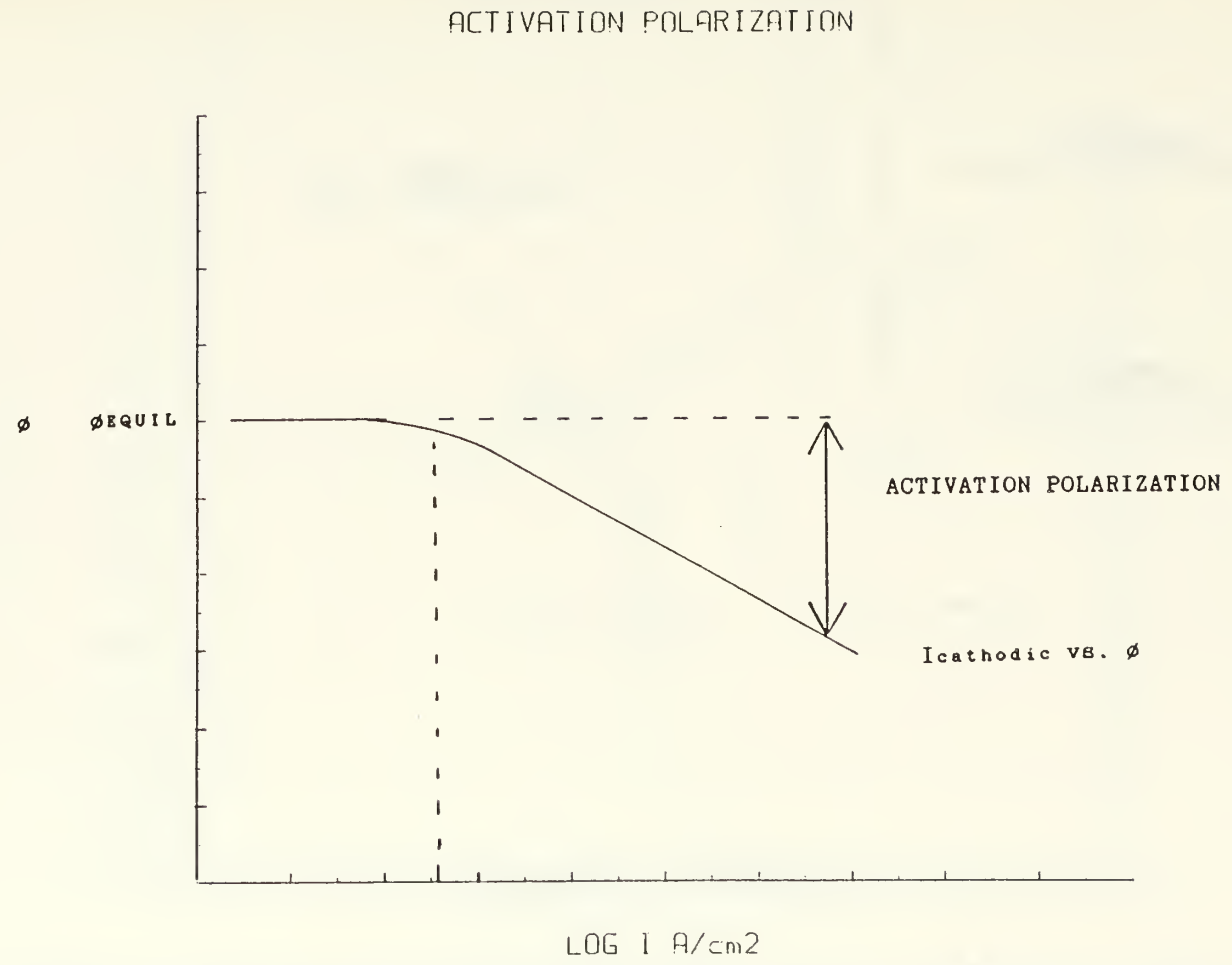

Figure 1. Theoretical activation polarization curve.

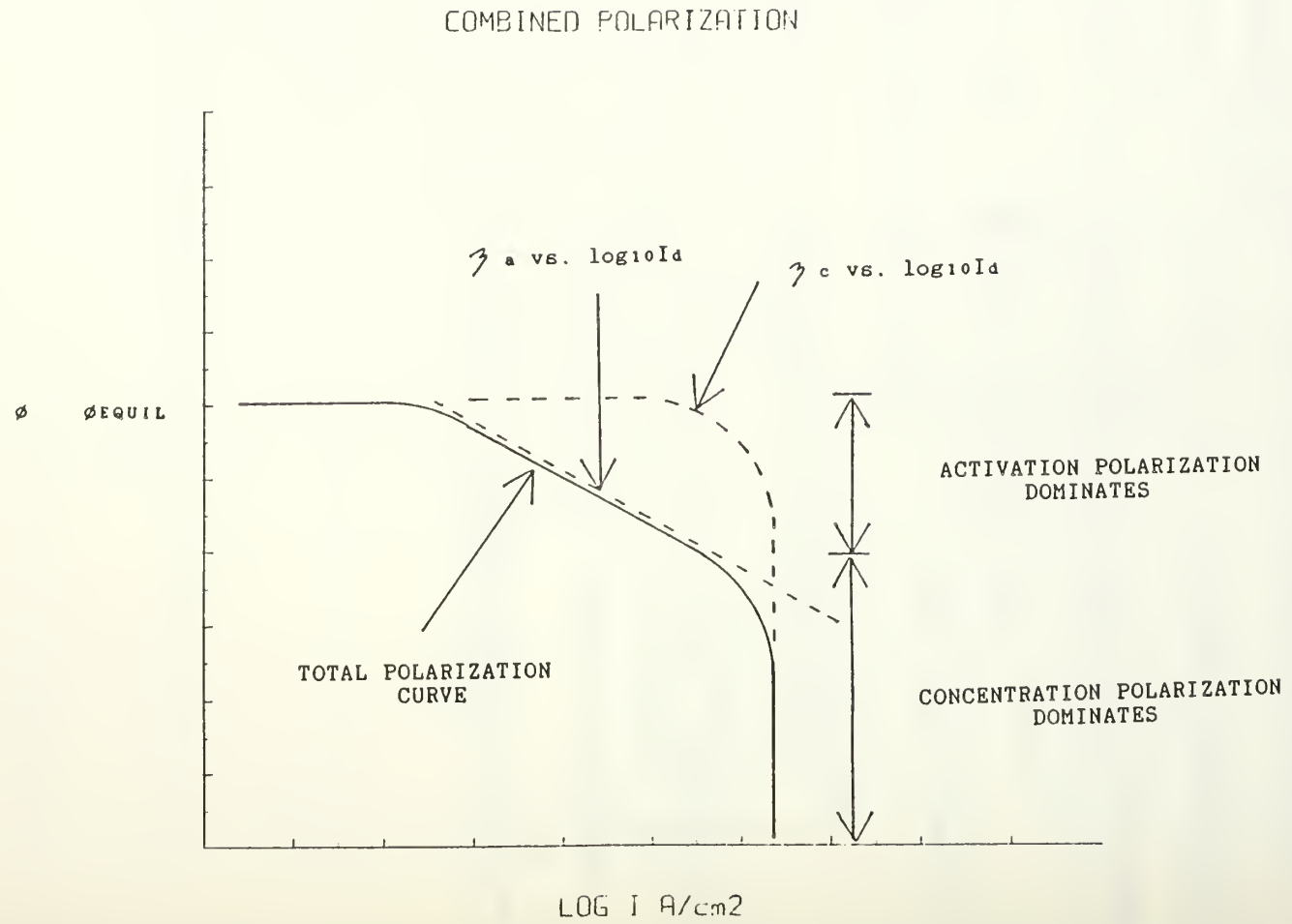

Figure 2. Theoretical activation and concentration polarization curve.

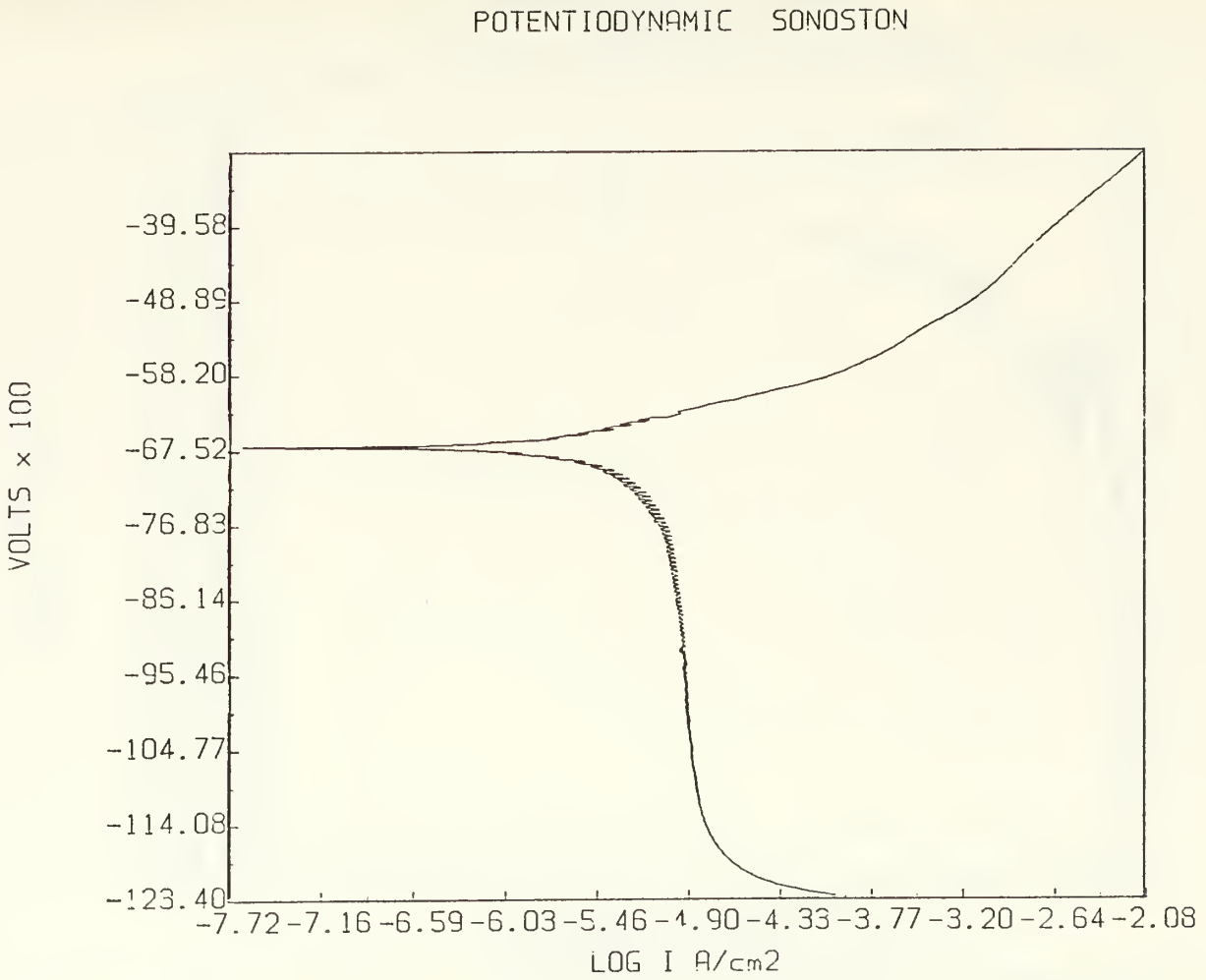

Figure 3. PDP plot of Cu-Mn-Al-Fe-Ni alloy in 3.5% NaCl solution.

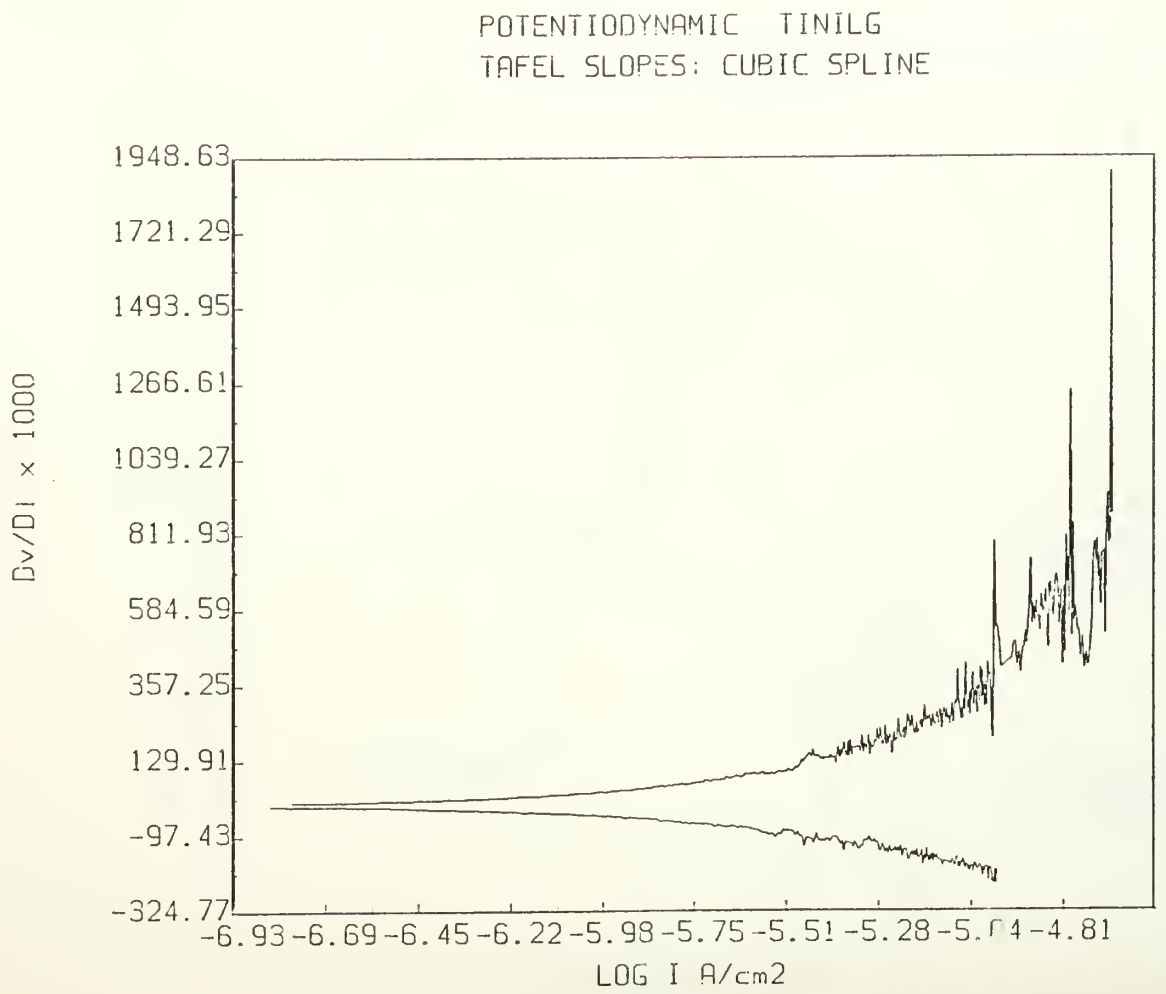

Figure 4. Cubic Spline derivatives of Ti-50% Ni alloy in synthetic seawater.

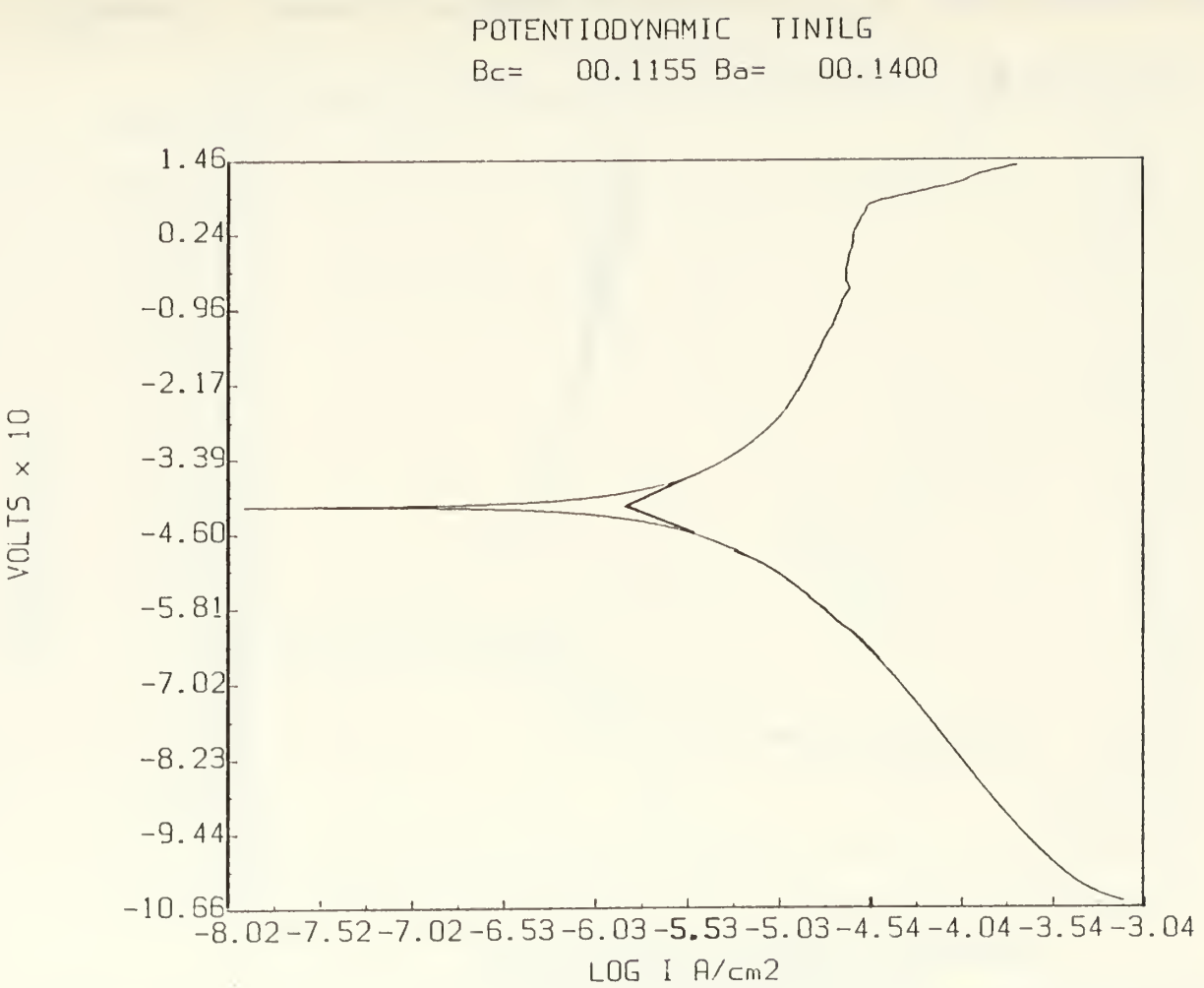

Figure 5. Tafel lines overlaid on PDP plot of Ti-50% Ni alloy in synthetic seawater.

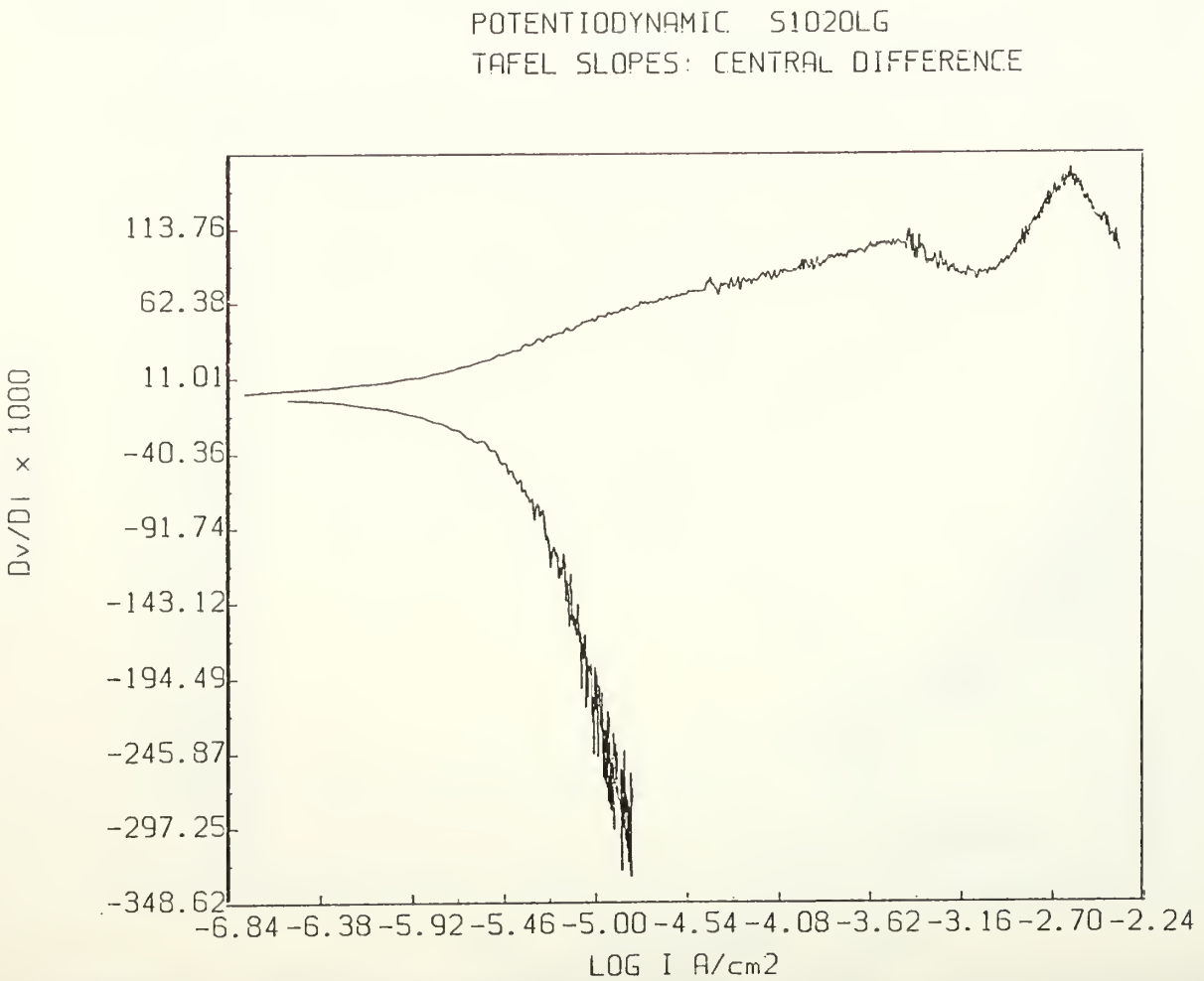

Figure 6. Central Difference derivatives of 1020 carbon steel in synthetic seawater.

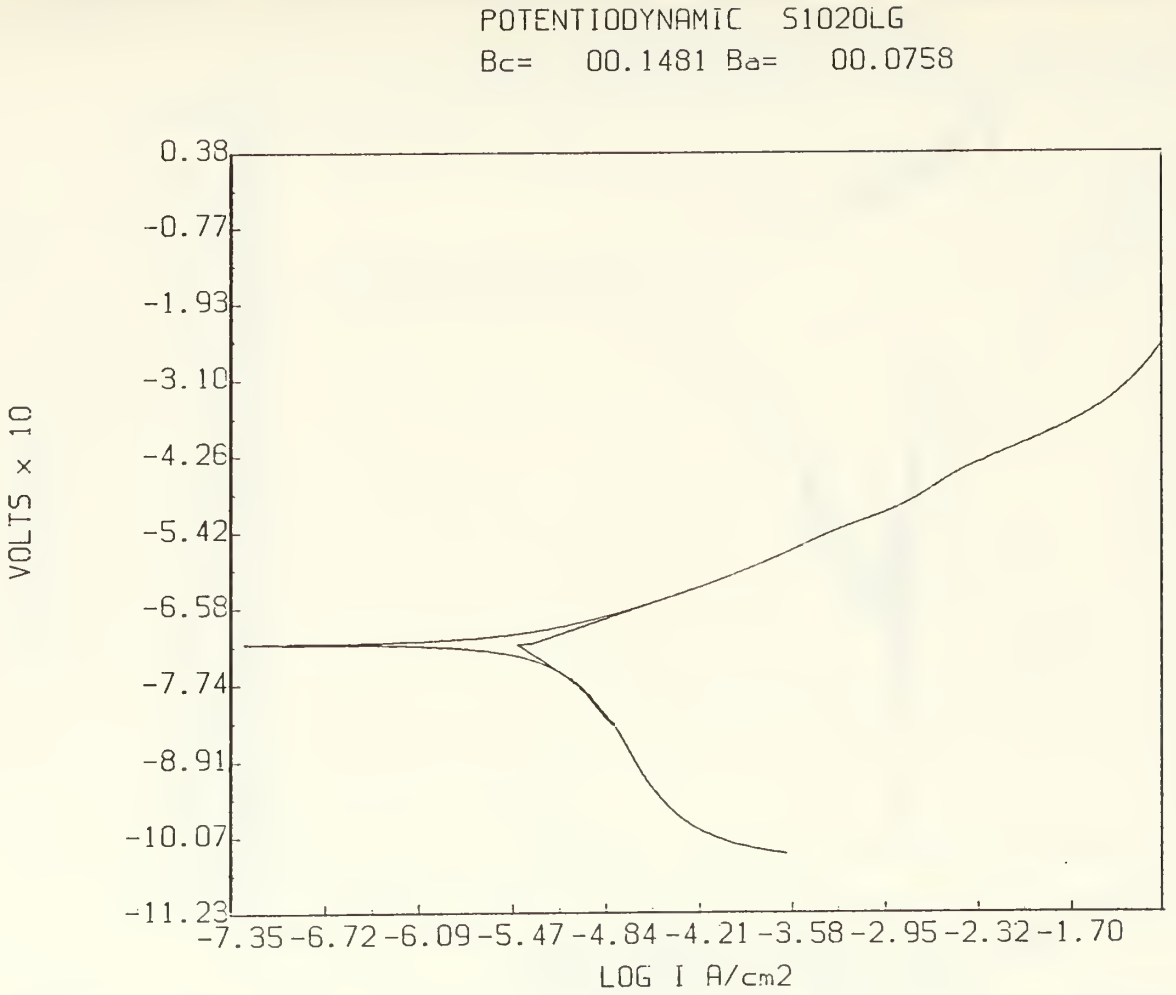

Figure 7. Tafel lines overlaid on PDP plot of 1020 carbon steel in synthetic seawater.

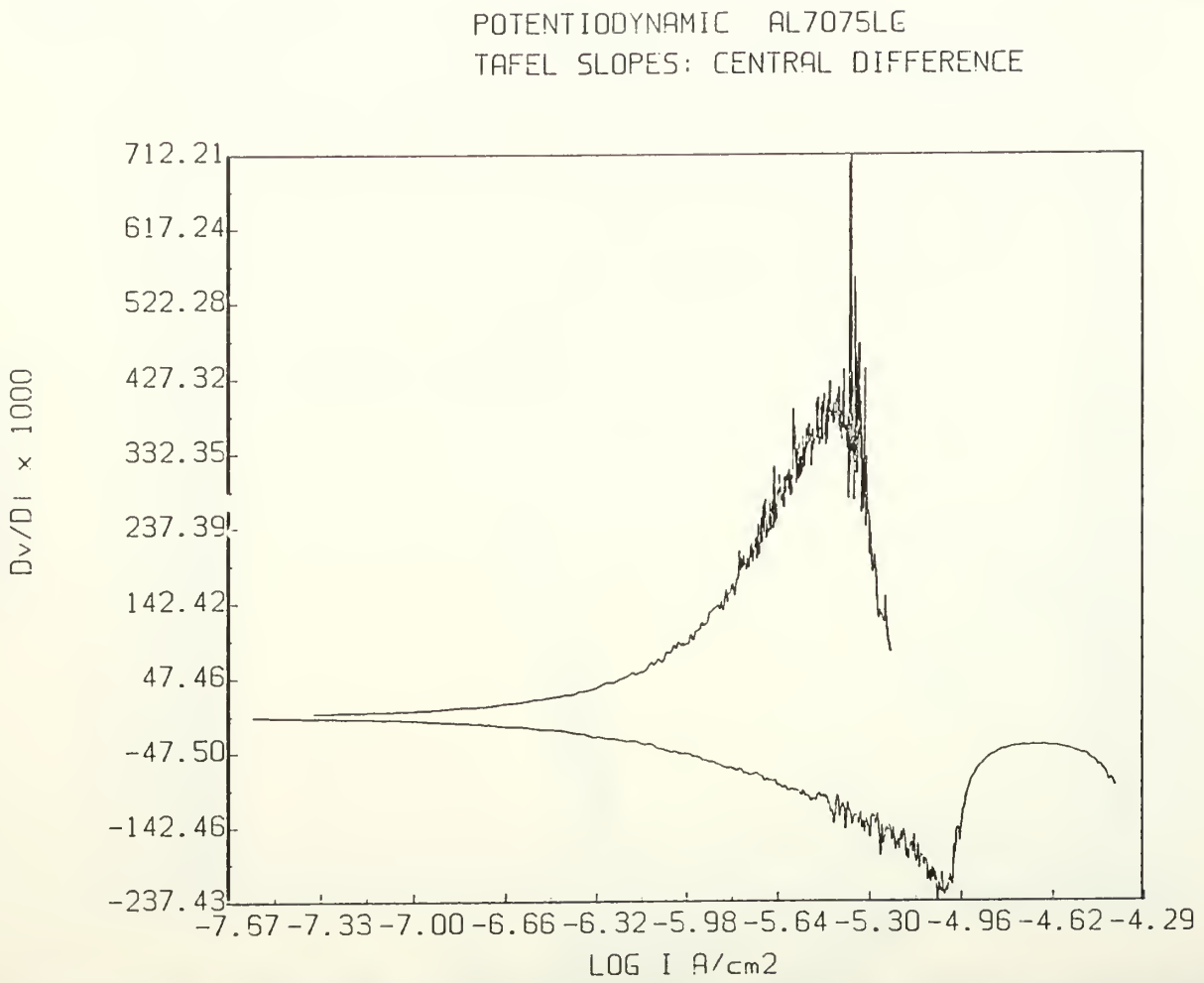

Figure 8. Central Difference derivatives of 7075 Aluminum alloy in synthetic seawater.

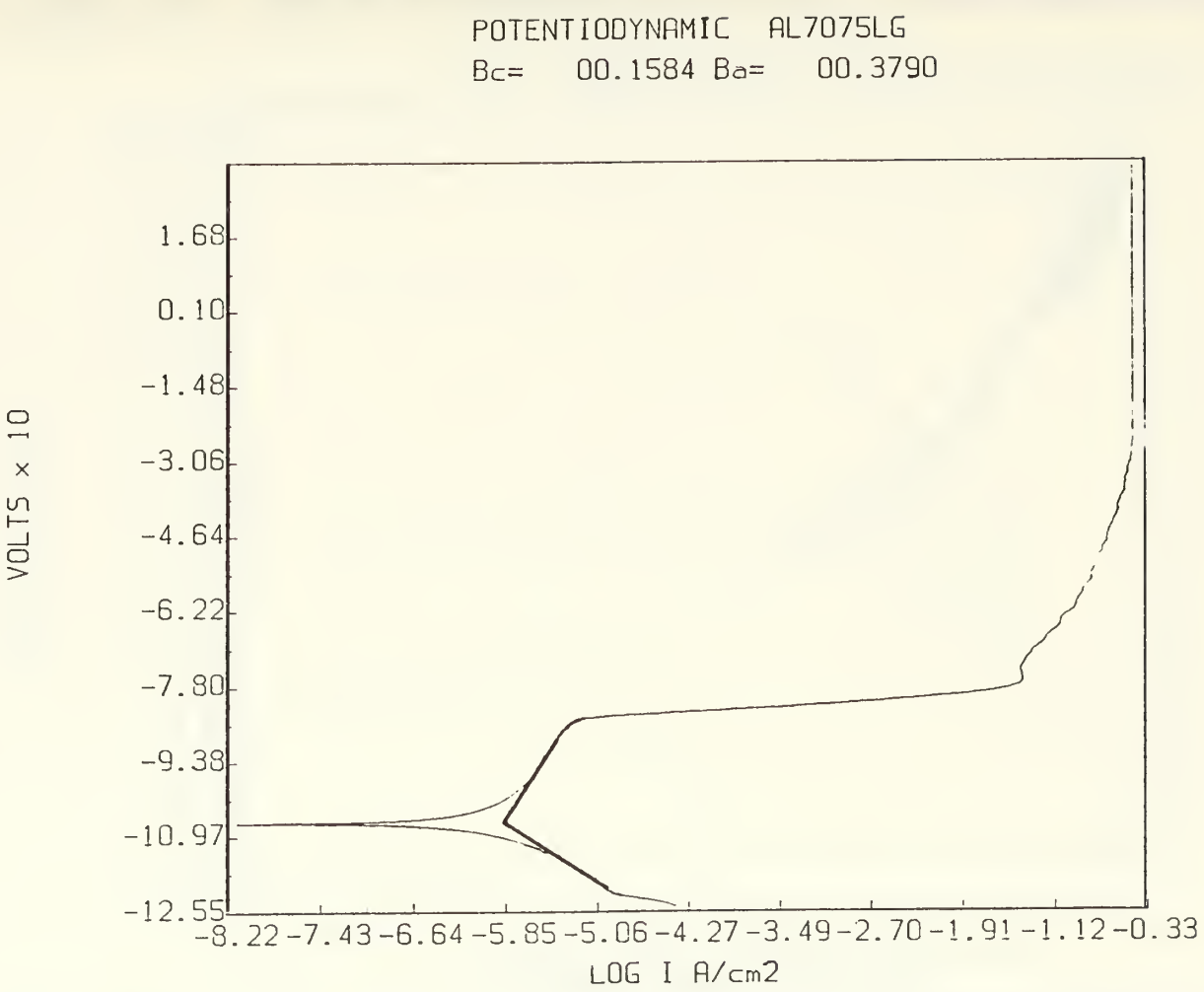

Figure 9. Tafel lines overlaid on PDP plot of 7075 Aluminum alloy in synthetic seawater.

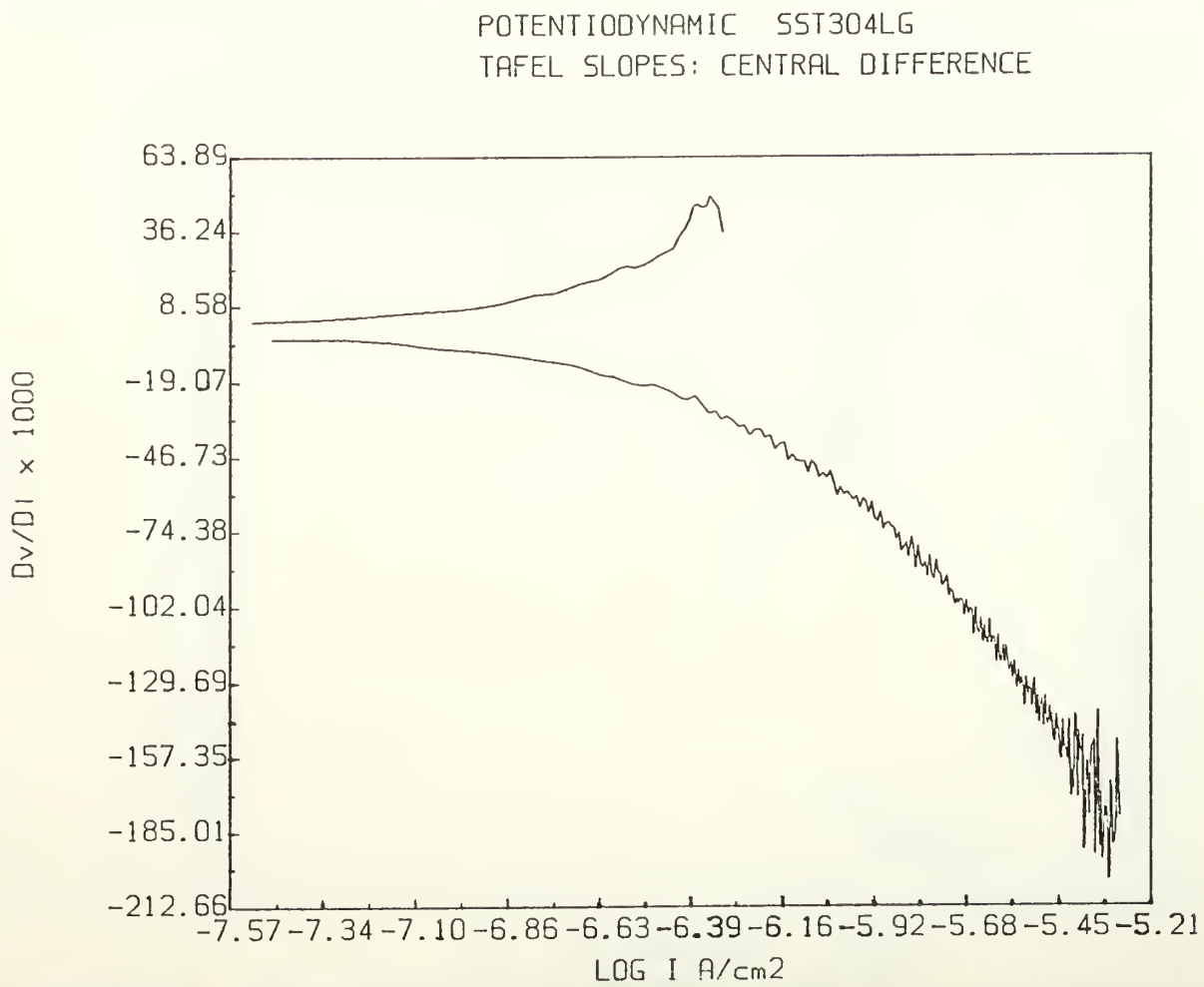

Figure 10. Central Difference derivatives of 304 Stainless steel in synthetic seawater.

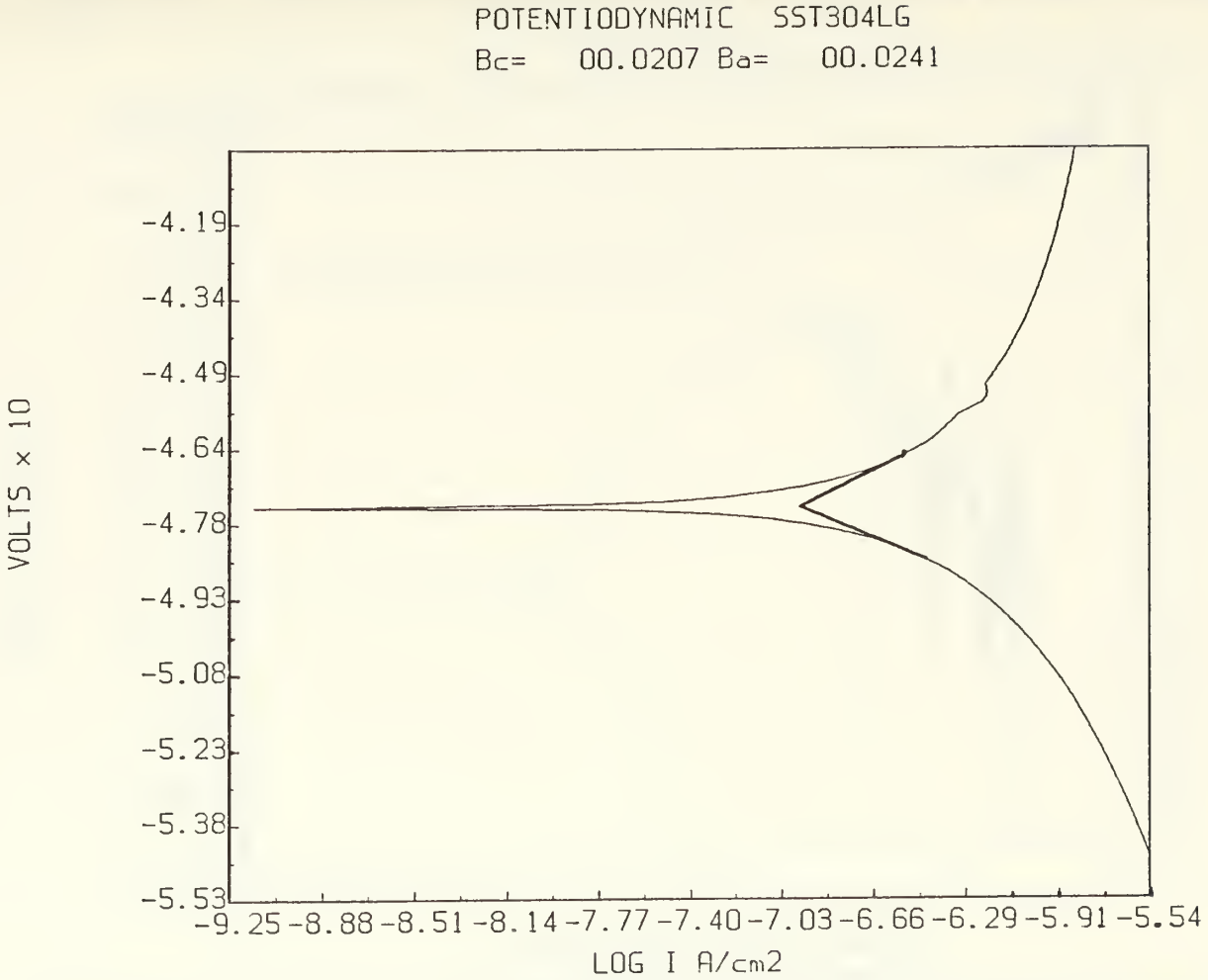

Figure 11. Tafel lines overlaid on PDP plot of 304 Stainless steel in synthetic seawater.

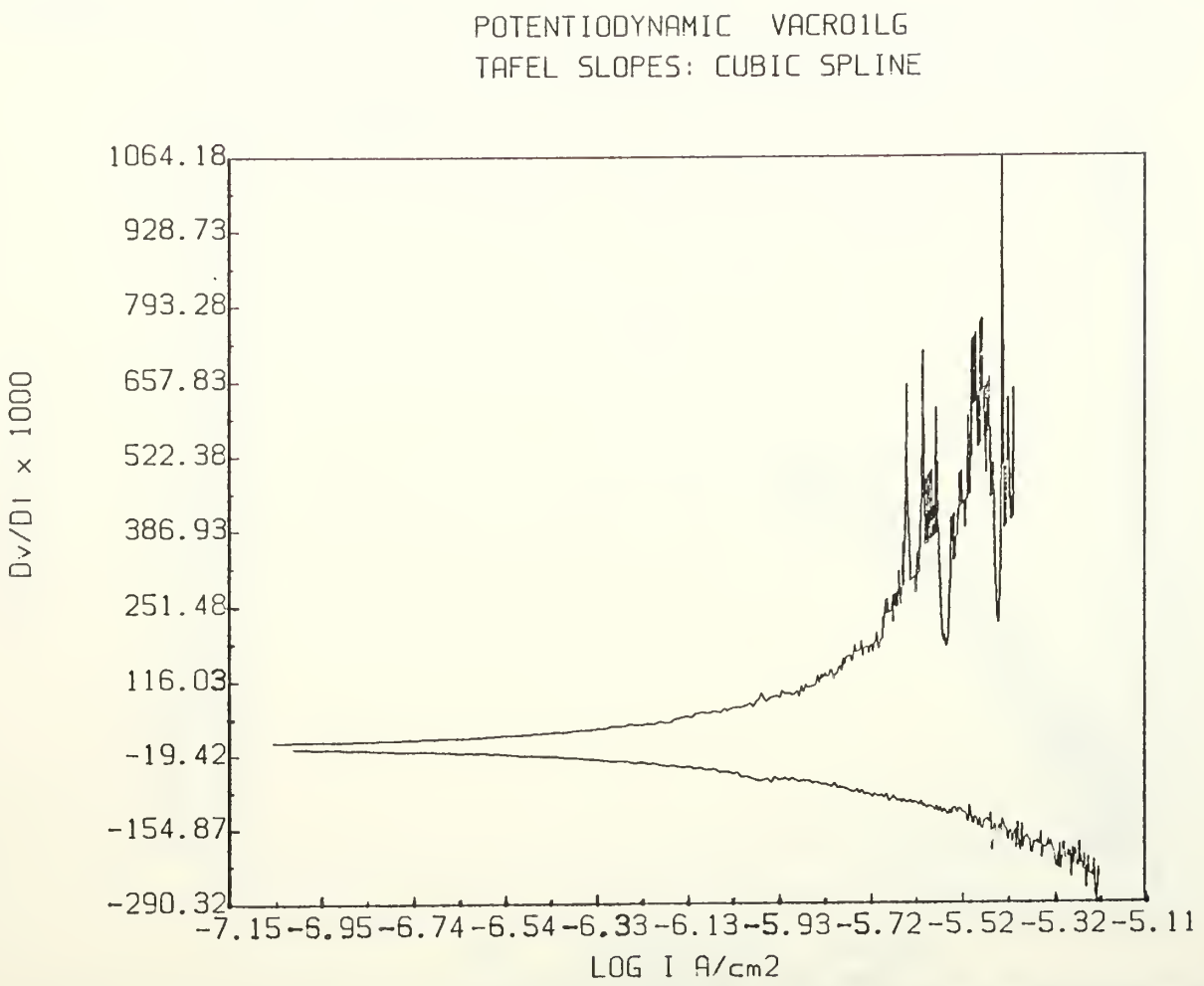

Figure 12. Cubic Spline derivatives of Fe-Cr-Mo alloy in synthetic seawater.

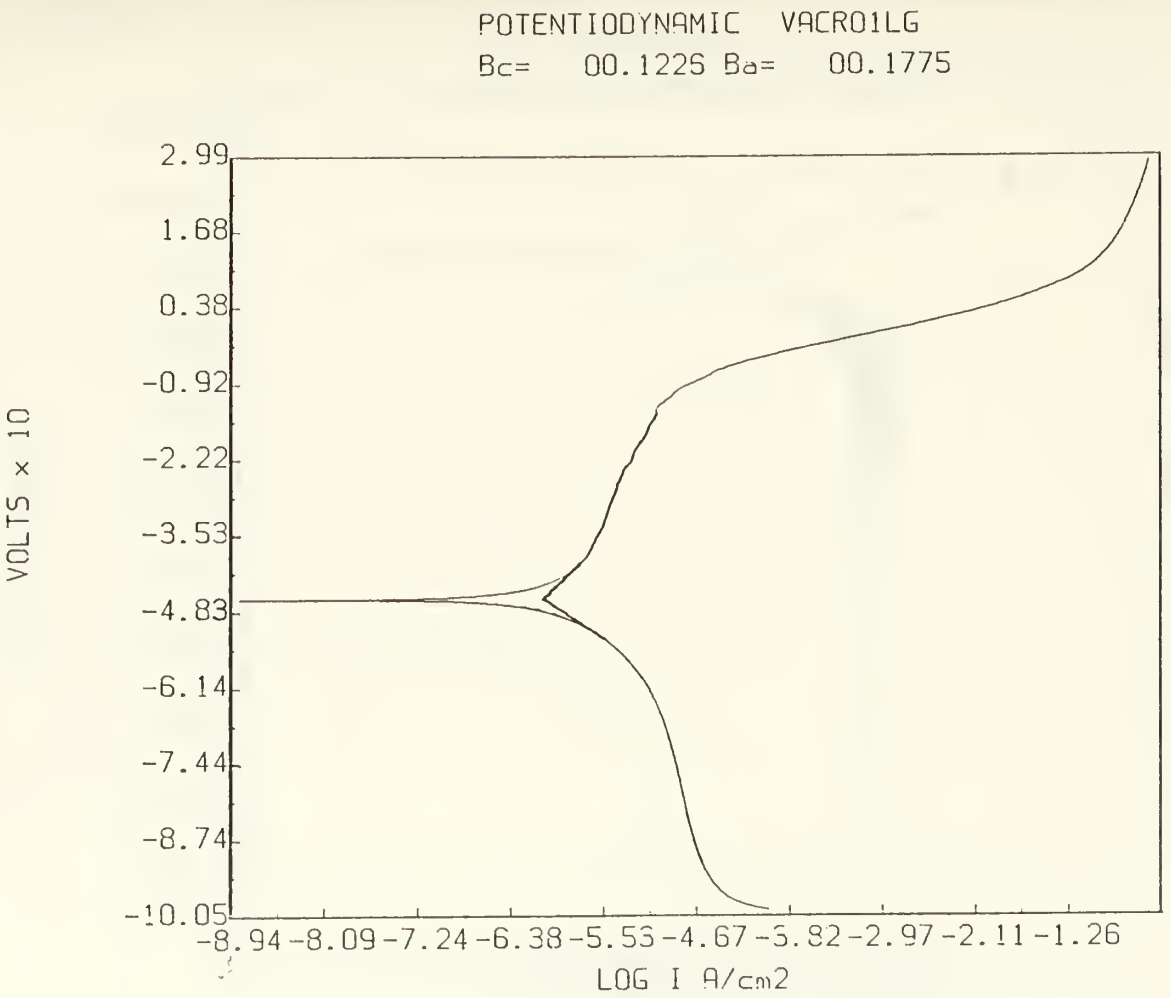

Figure 13. Tafel lines overlaid on PDP plot of Fe-Cr-Mo alloy in synthetic seawater.

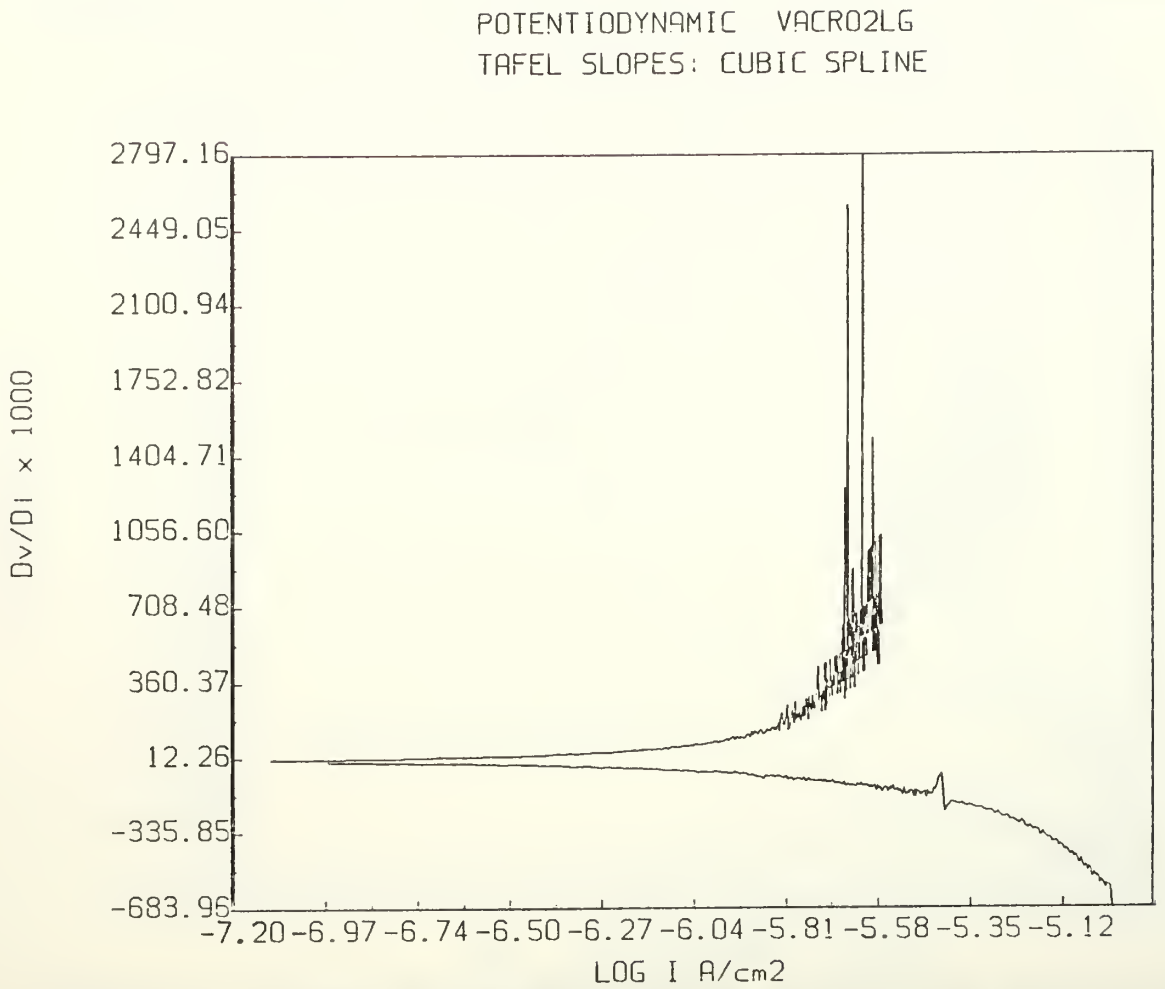

Figure 14. Cubic Spline derivatives of Fe-Cr-Al alloy in synthetic seawater.

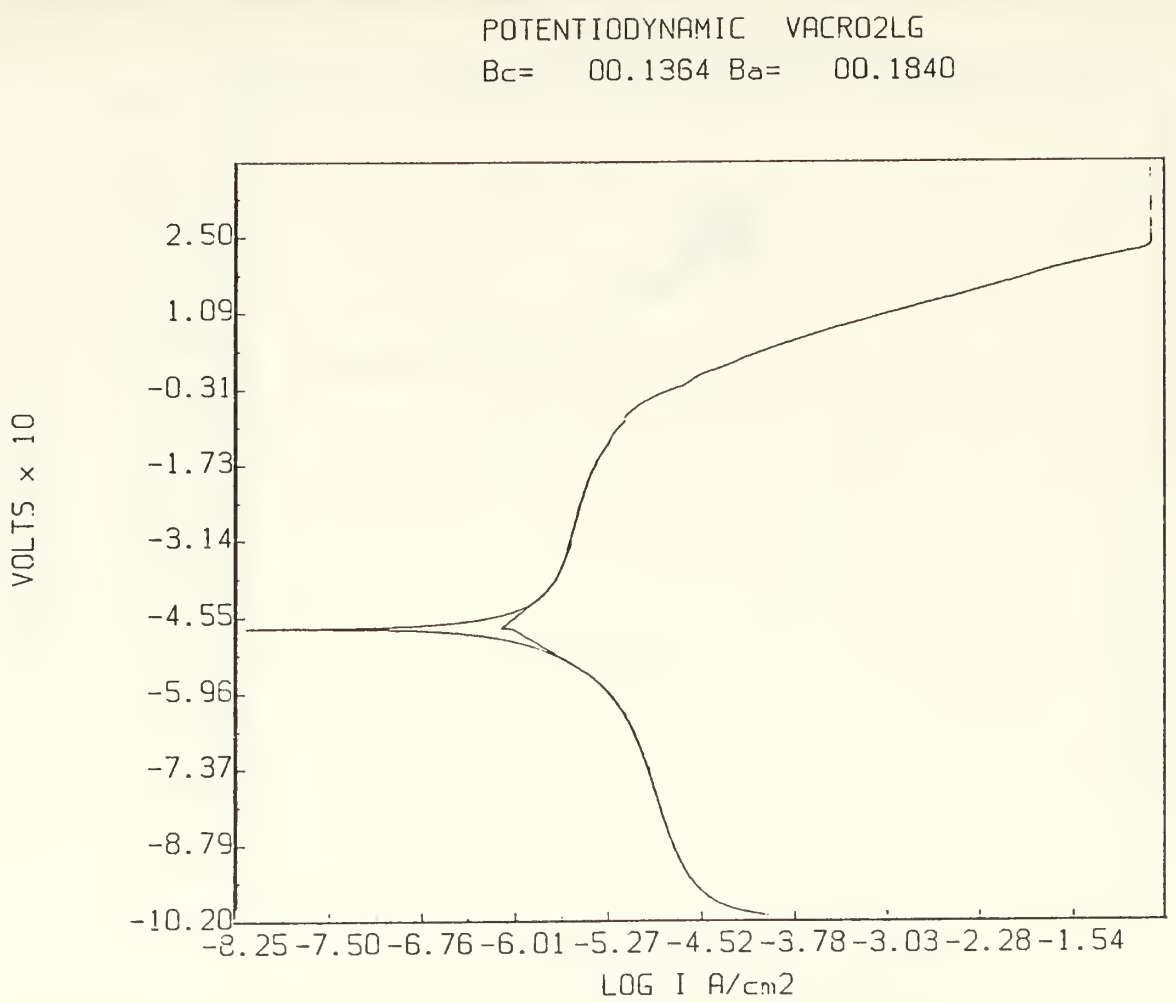

Figure 15. Tafel lines overlaid on PDP plot of Fe-Cr-Al alloy in synthetic seawater.

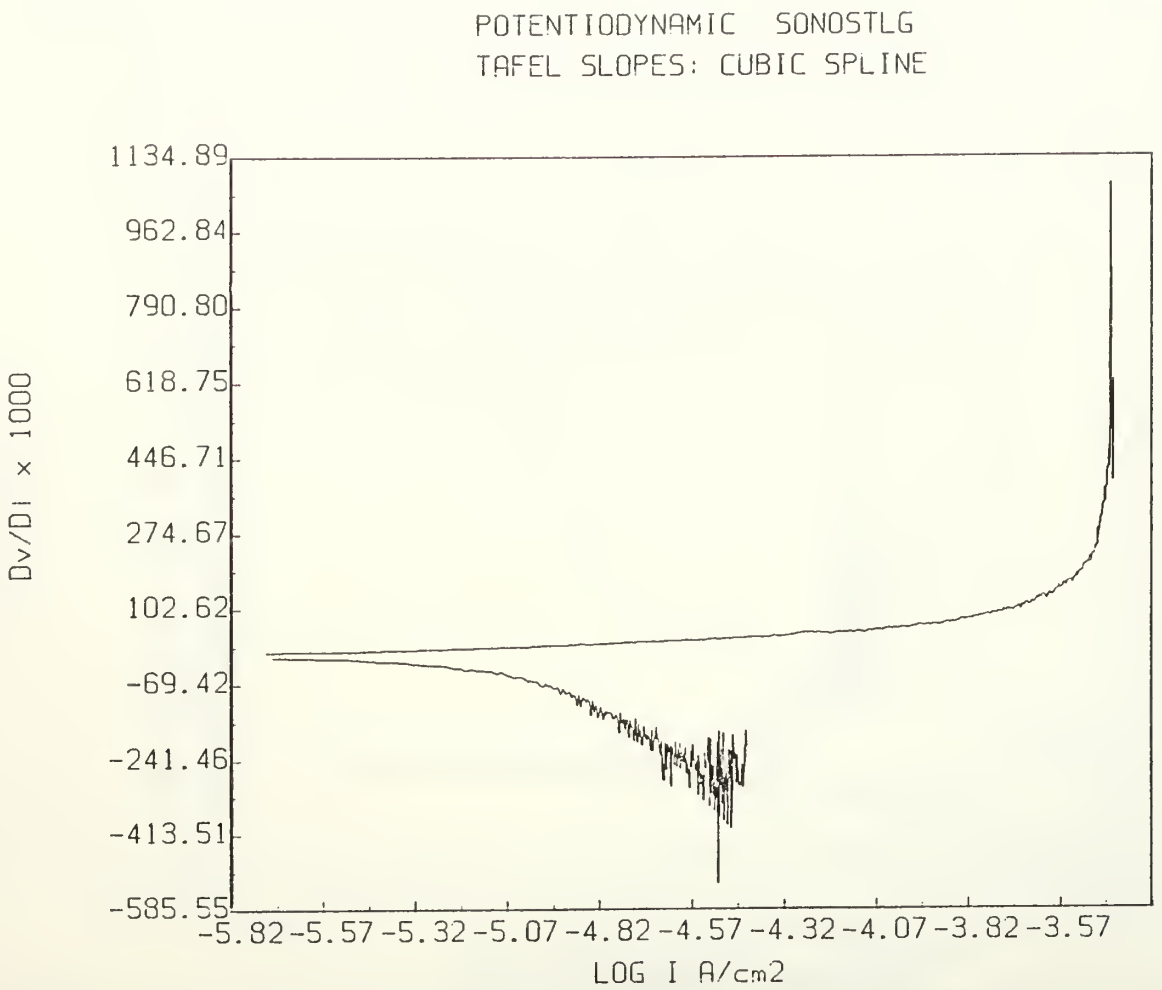

Figure 16. Cubic Spline Derivatives of Cu-Mn-Al-Fe-Ni alloy in synthetic seawater.

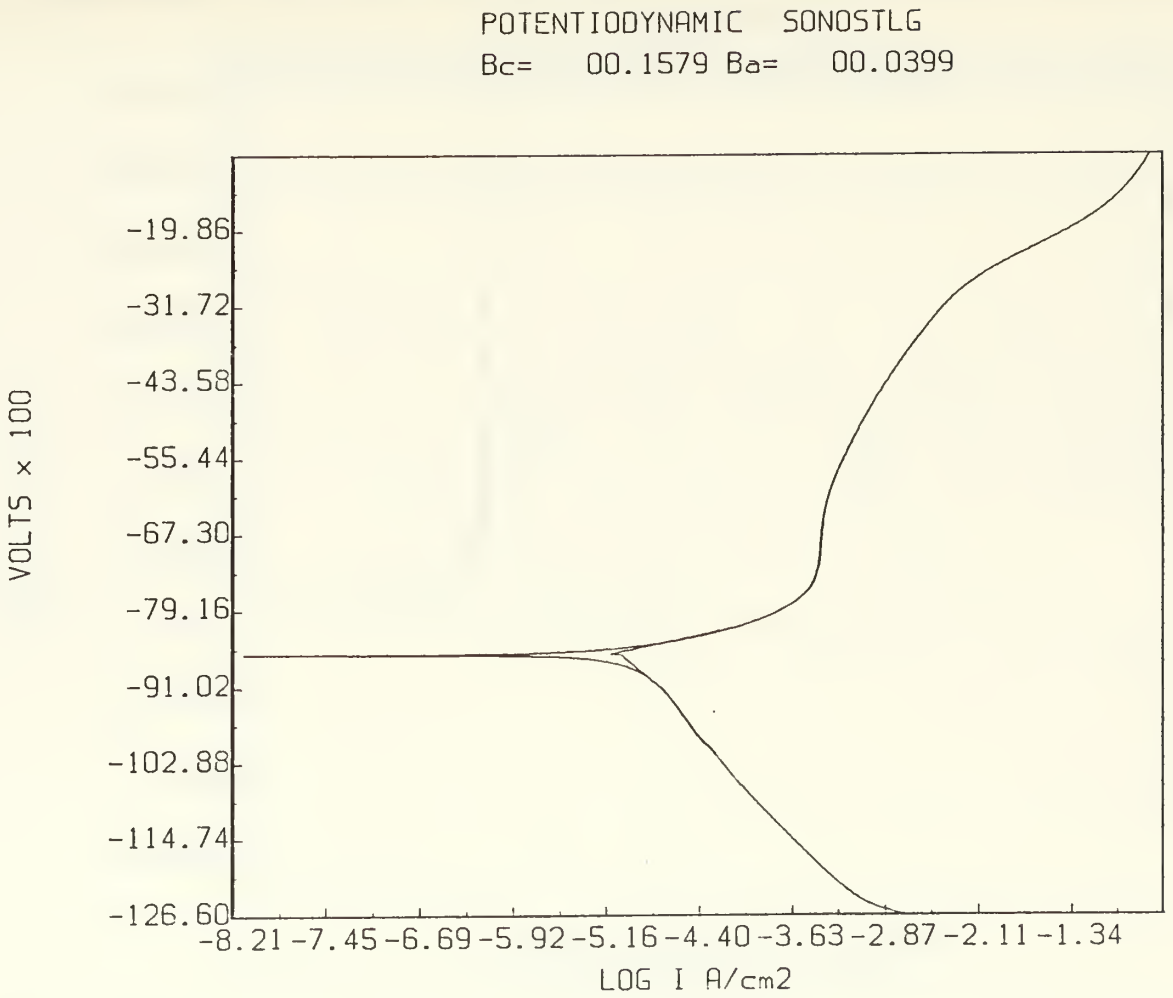

Figure 17. Tafel lines overlaid on PDP plot of Cu-Mn-Al-Fe-Ni alloy in synthetic seawater.

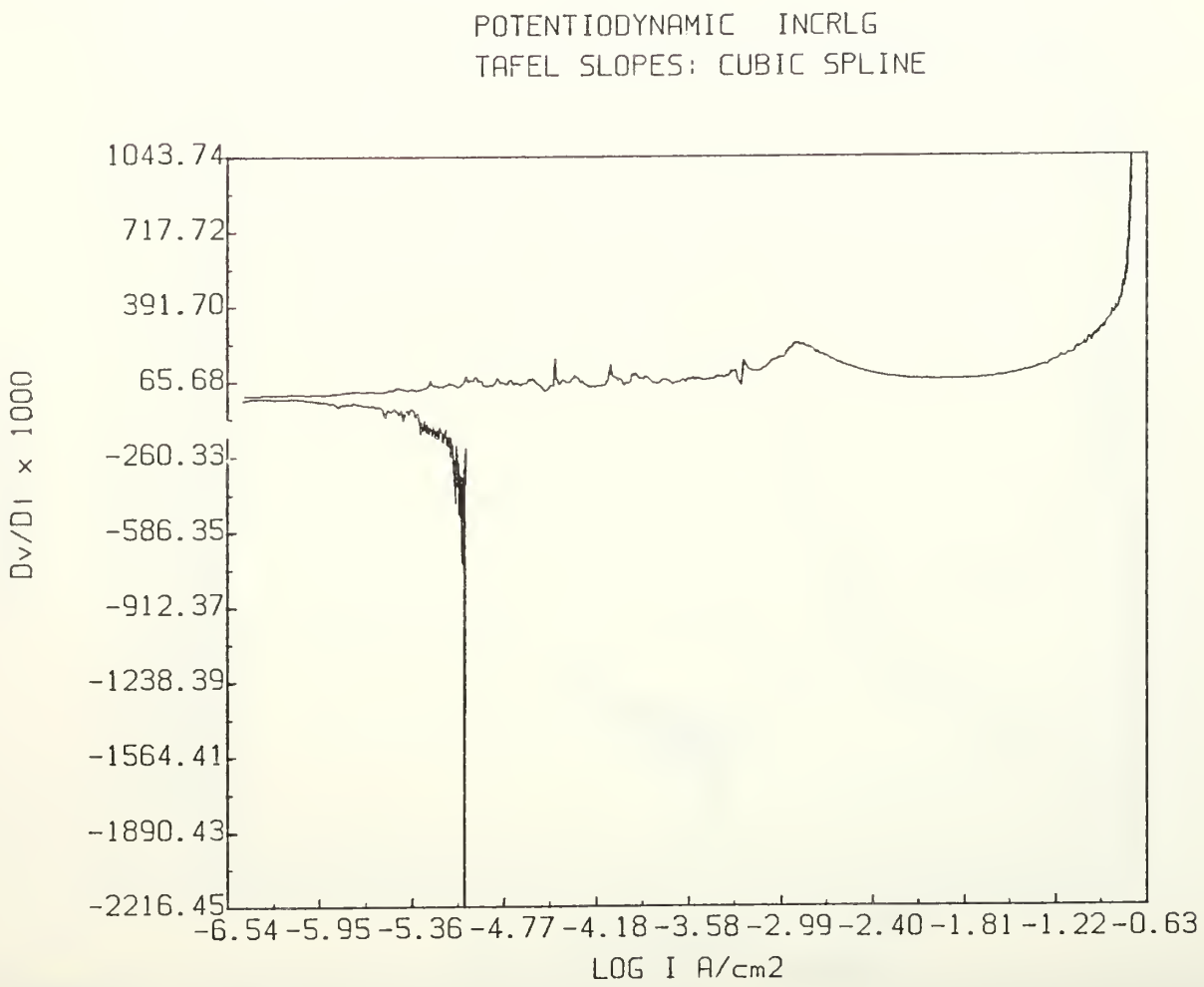

Figure 18. Cubic Spline derivatives of Cu-Mn-Al alloy in synthetic seawater.

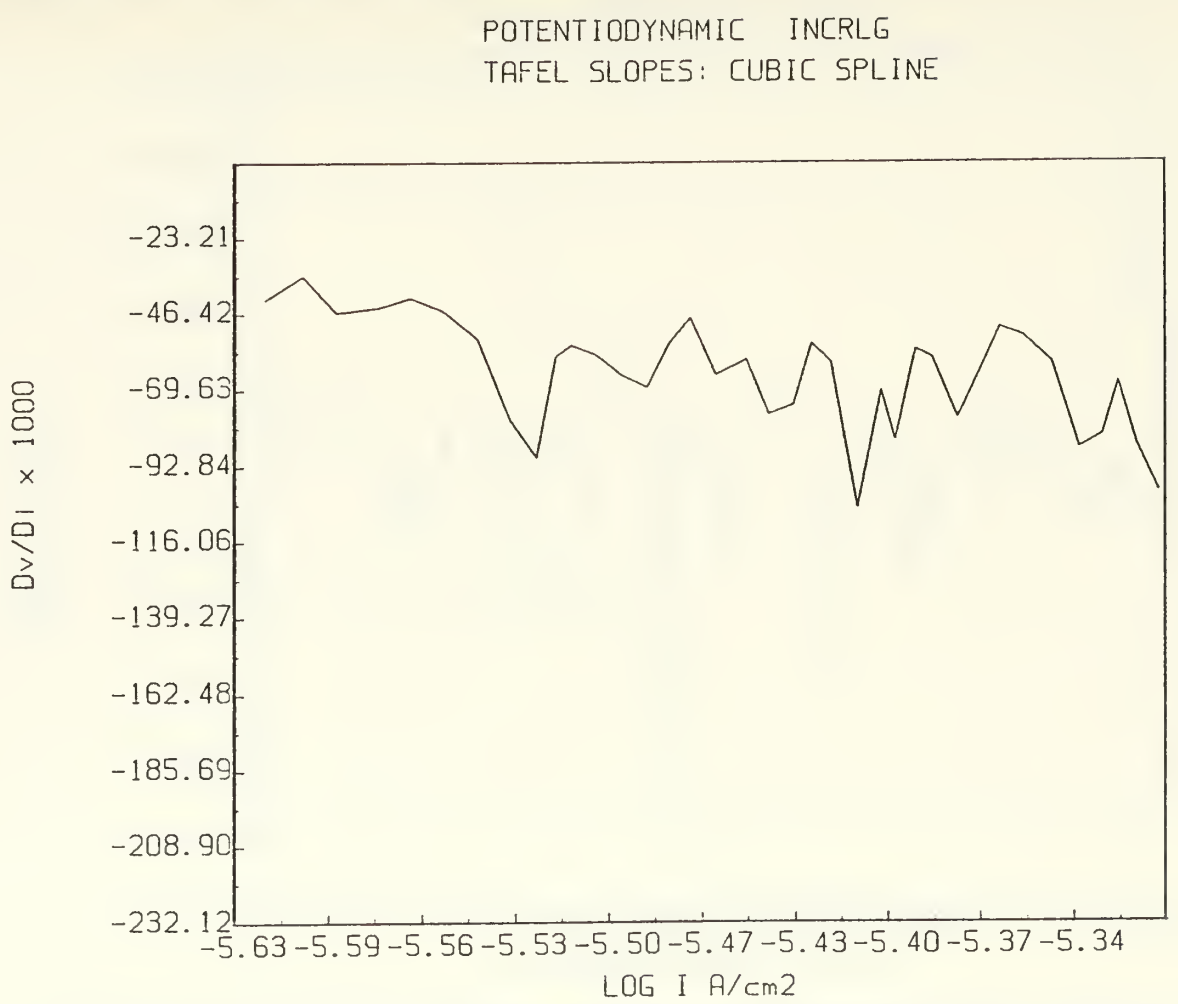

Figure 19. Cubic Spline derivatives of the cathodic branch of Cu-Mn-Al alloy in synthetic seawater.

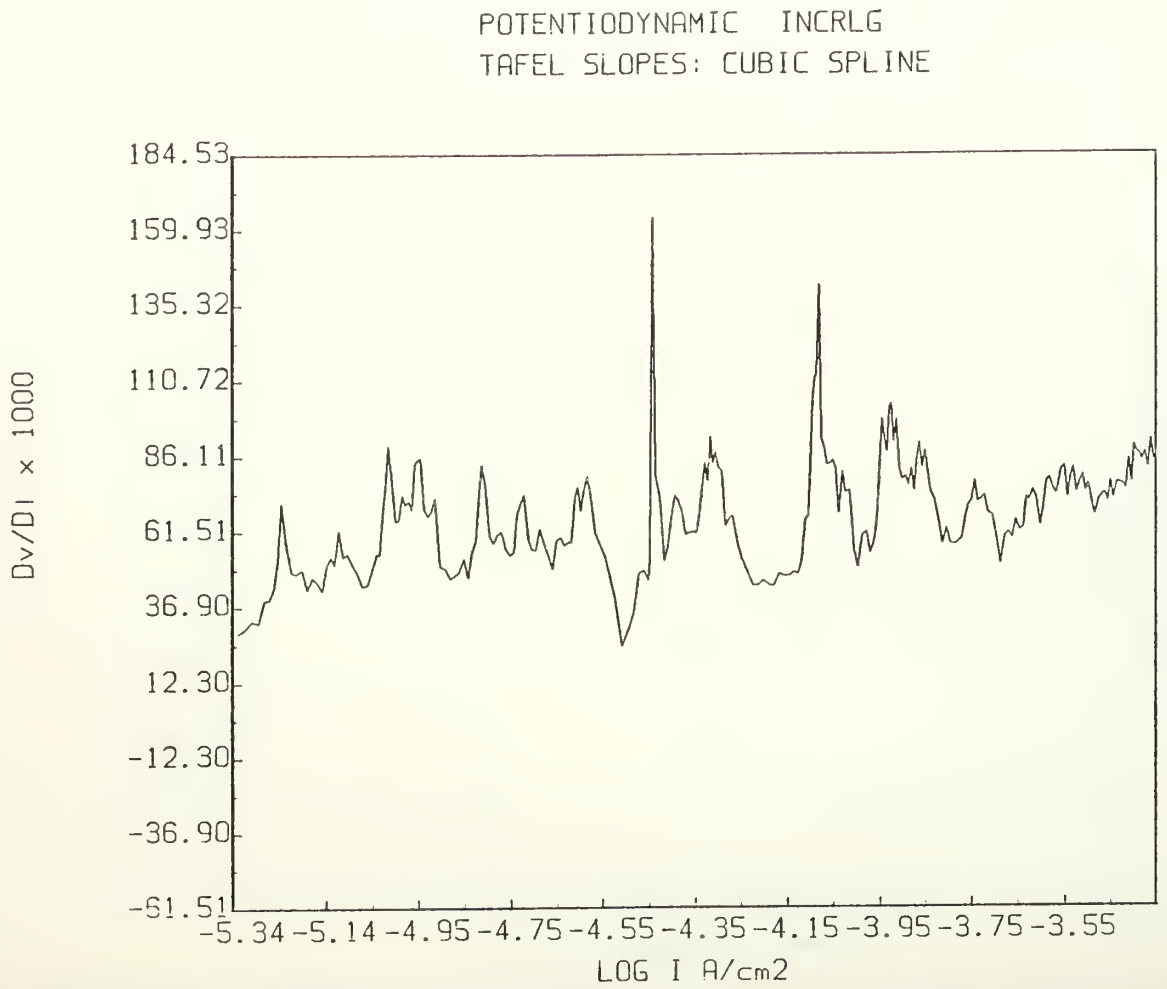

Figure 20. Cubic Spline derivatives of the anodic branch of Cu-Mn-Al alloy in synthetic seawater.

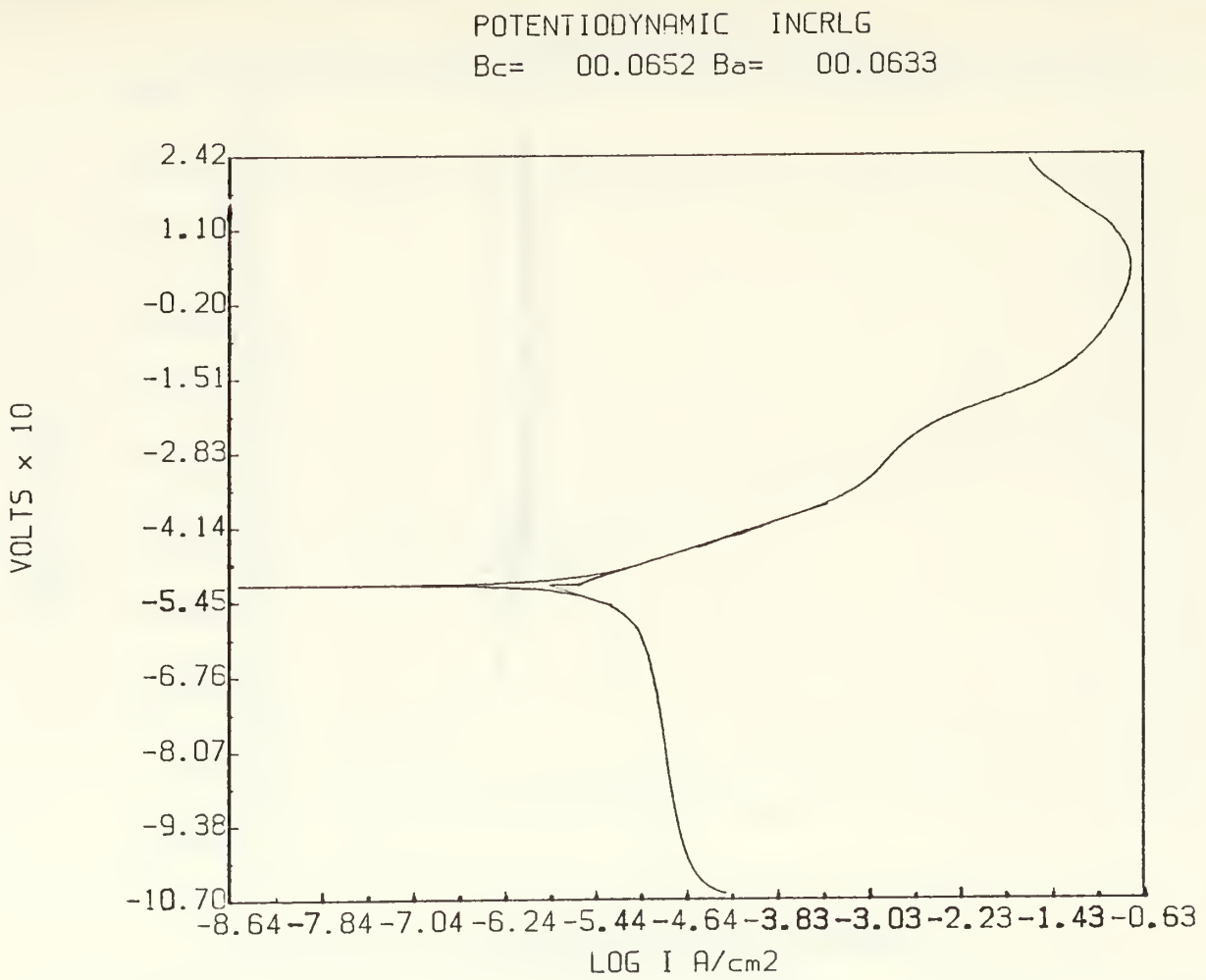

Figure 21. Tafel lines overlaid on PDP plot of Cu-Mn-Al alloy in synthetic seawater.

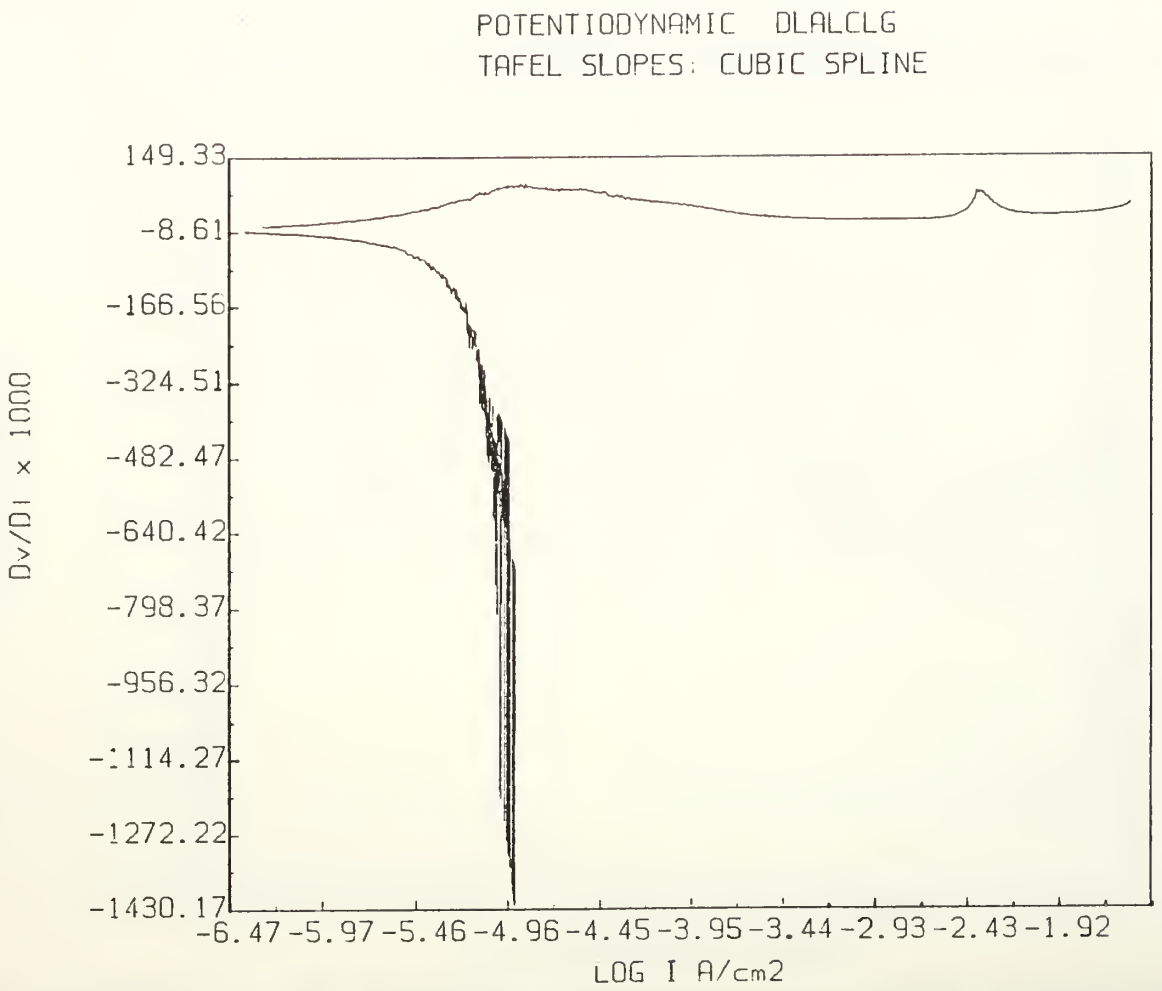

Figure 22. Cubic Spline derivatives of Cu-Zn-Al alloy in synthetic seawater.

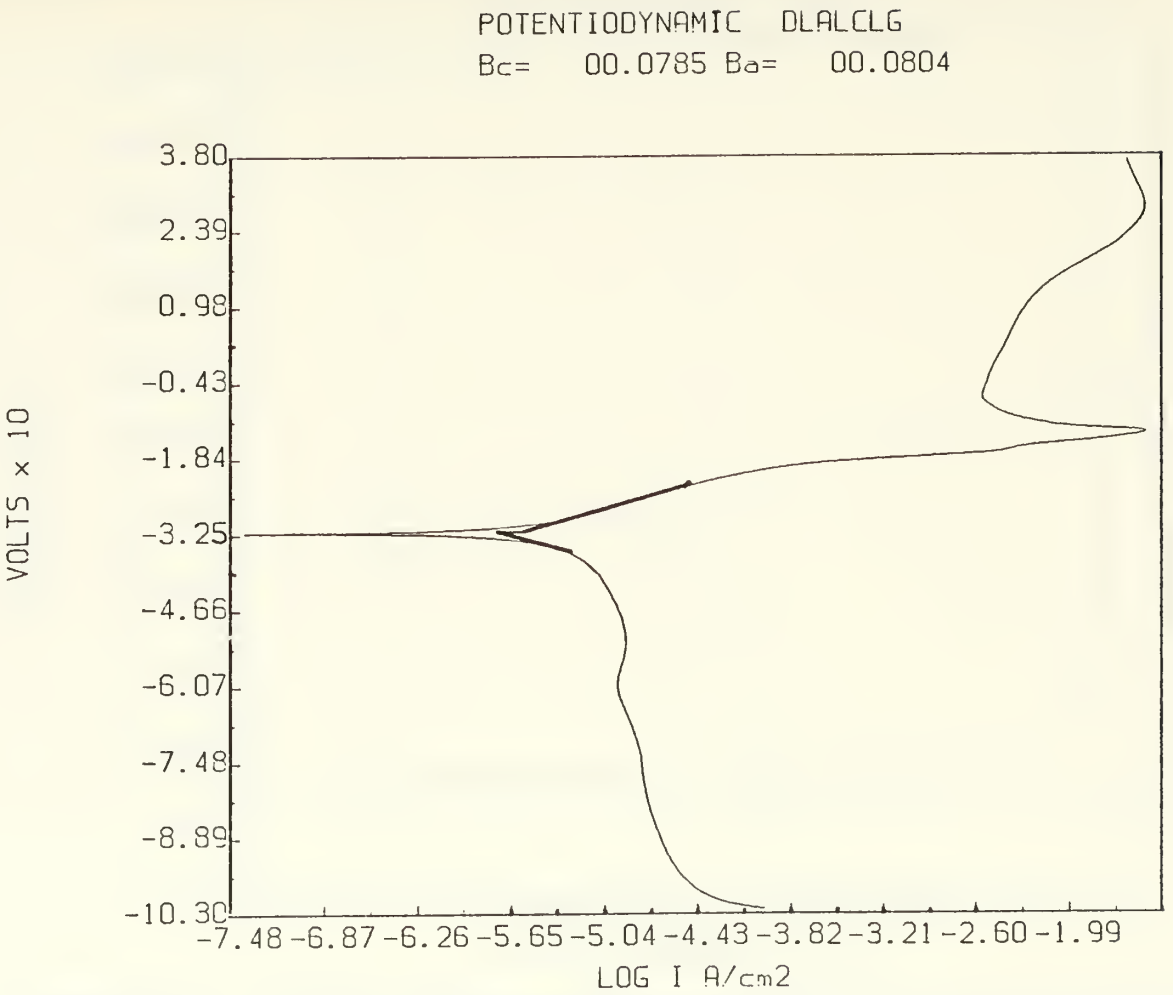

Figure 23. Tafel lines overlaid on PDP plot of Cu-Zn-Al alloy in synthetic seawater.

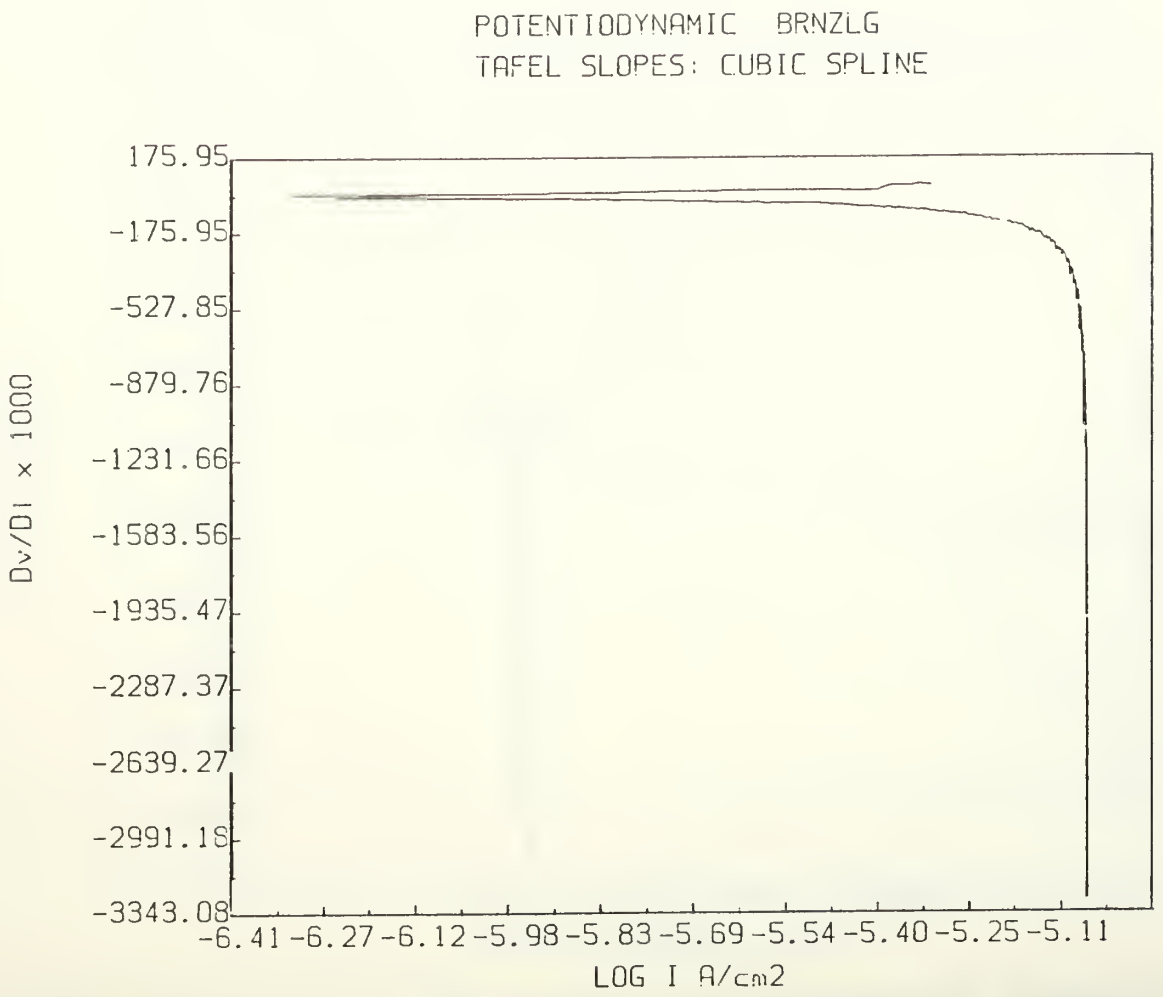

Figure 24. Cubic Spline derivatives of 630 series Bronze in synthetic seawater.

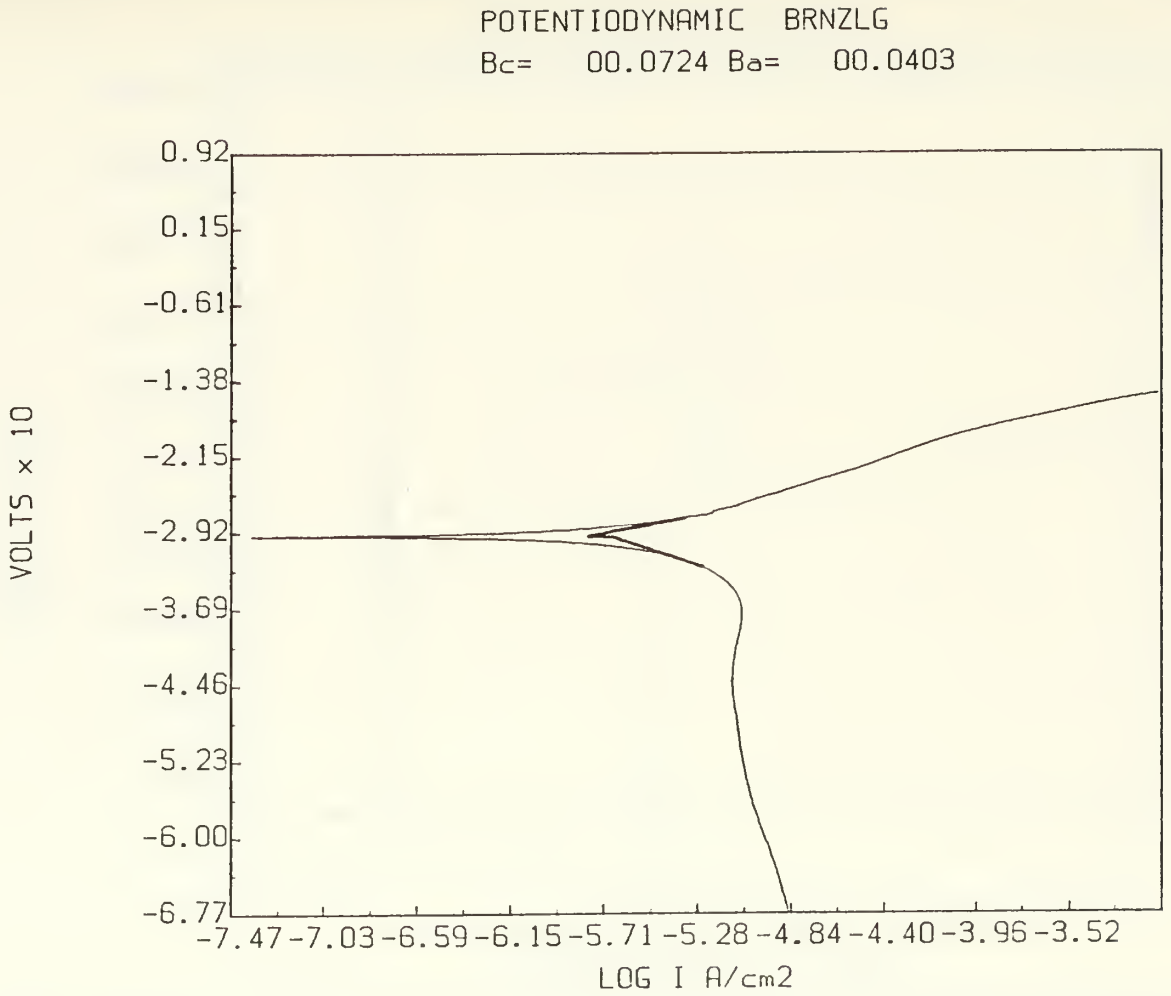

Figure 25. Tafel lines overlaid on PDP plot of 630 series Bronze in synthetic seawater.

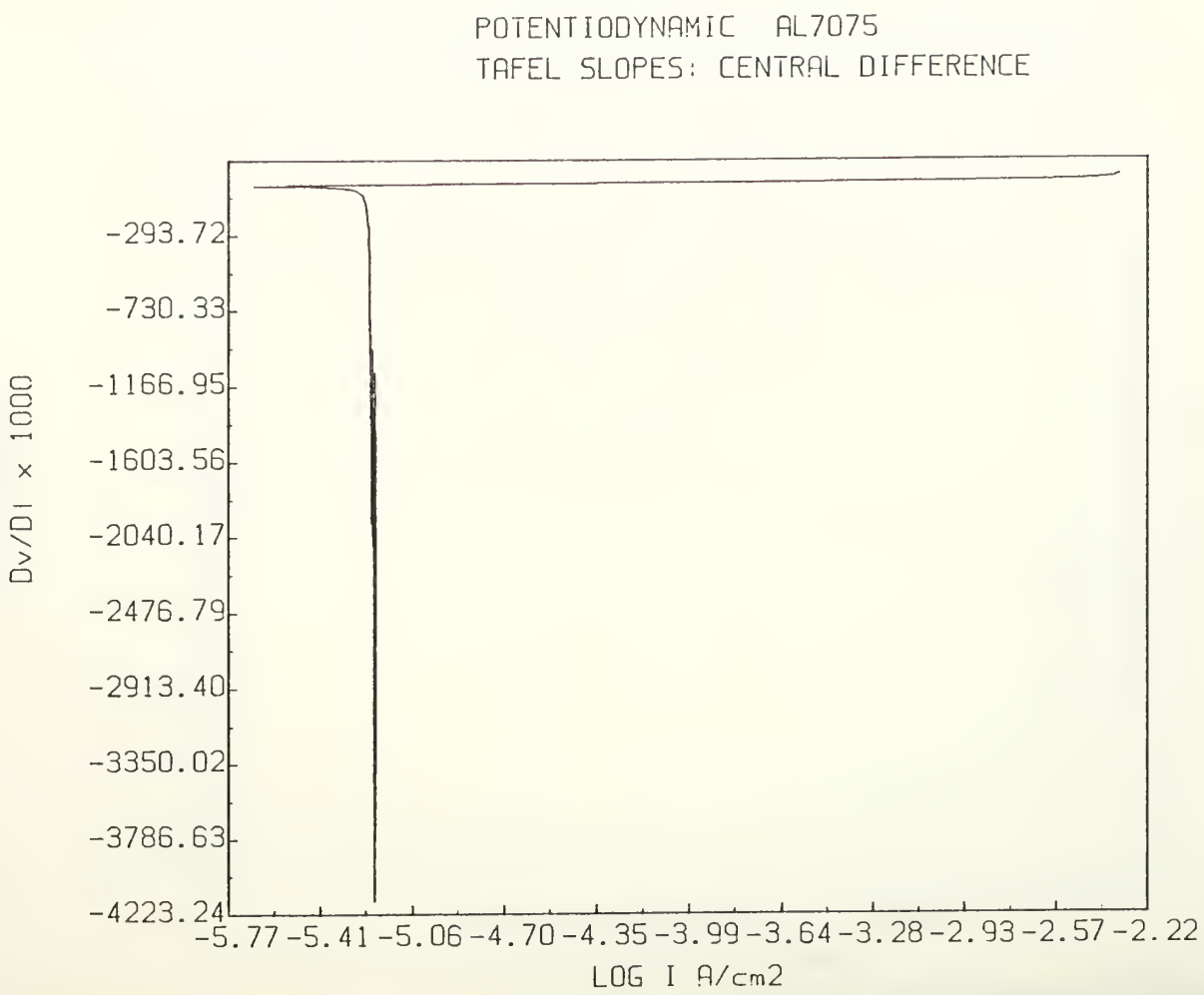

Figure 26. Central Difference derivatives of 7075 Aluminum alloy in 3.5% NaCl solution.

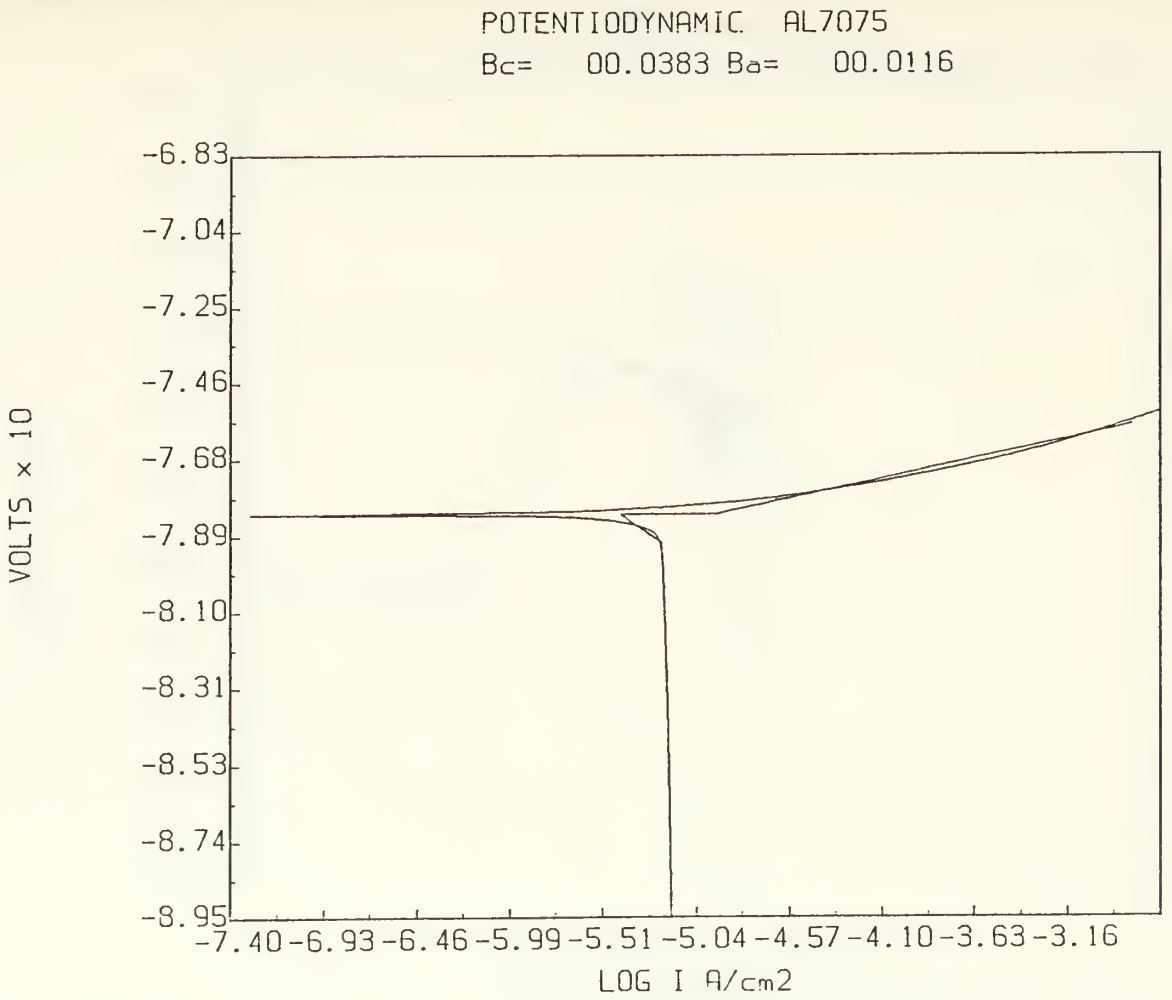

Figure 27. Tafel lines overlaid on PDP plot of 7075 Aluminum alloy in 3.5% NaCl solution.

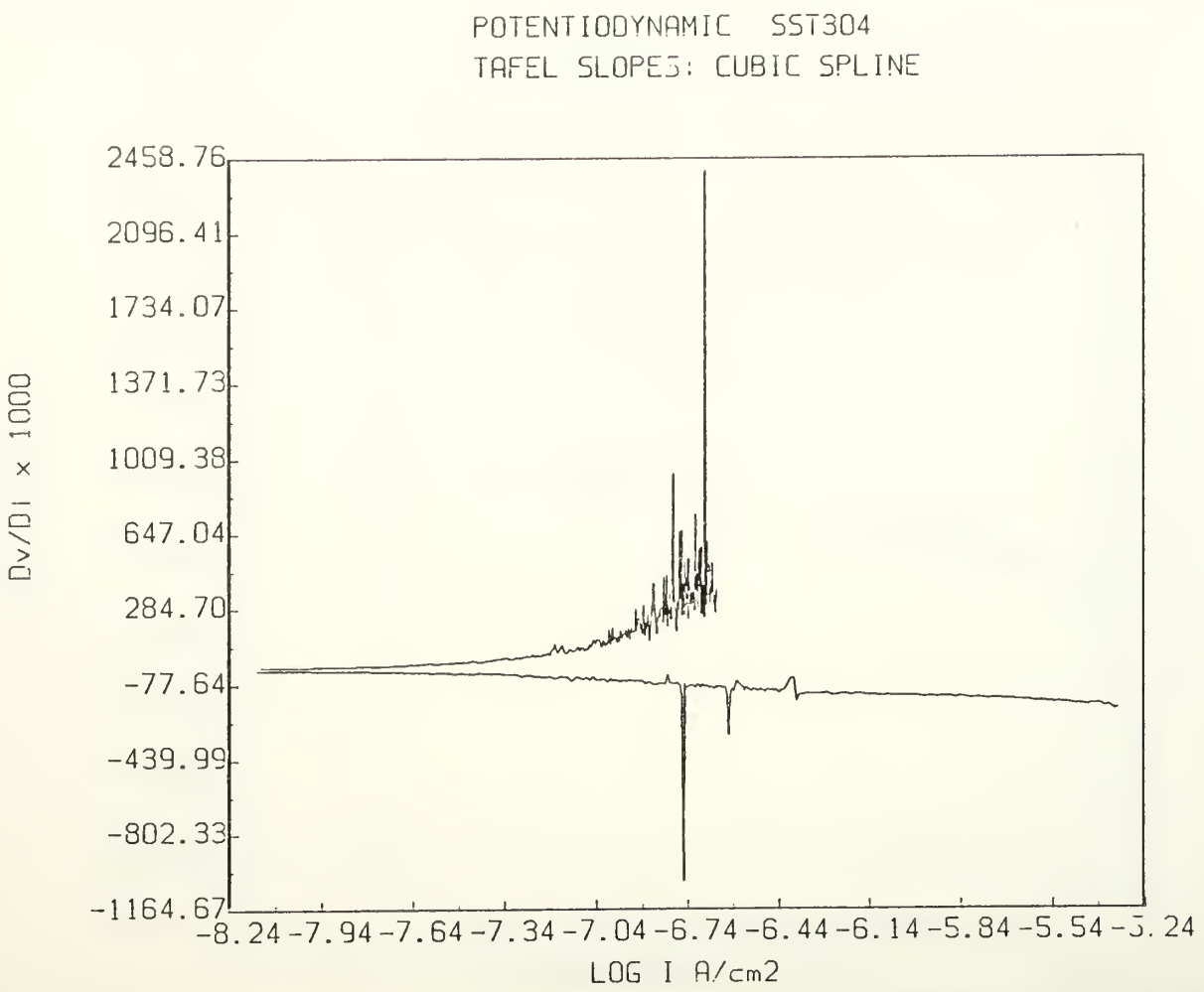

Figure 28. Cubic Spline derivatives of 304 Stainless steel in 3.5% NaCl solution.

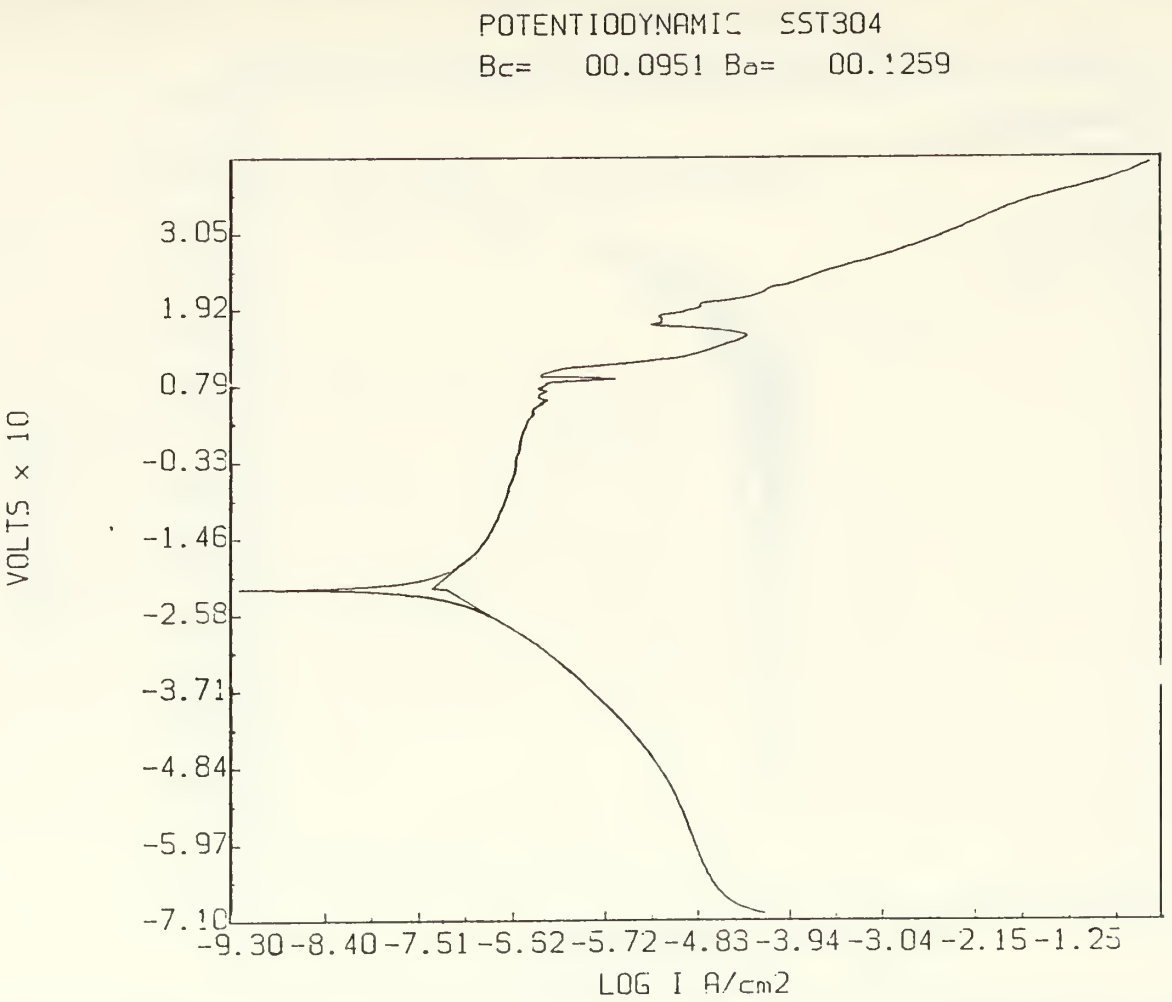

Figure 29. Tafel lines overlaid on PDP plot of 304 Stainless steel in 3.5% NaCl solution.

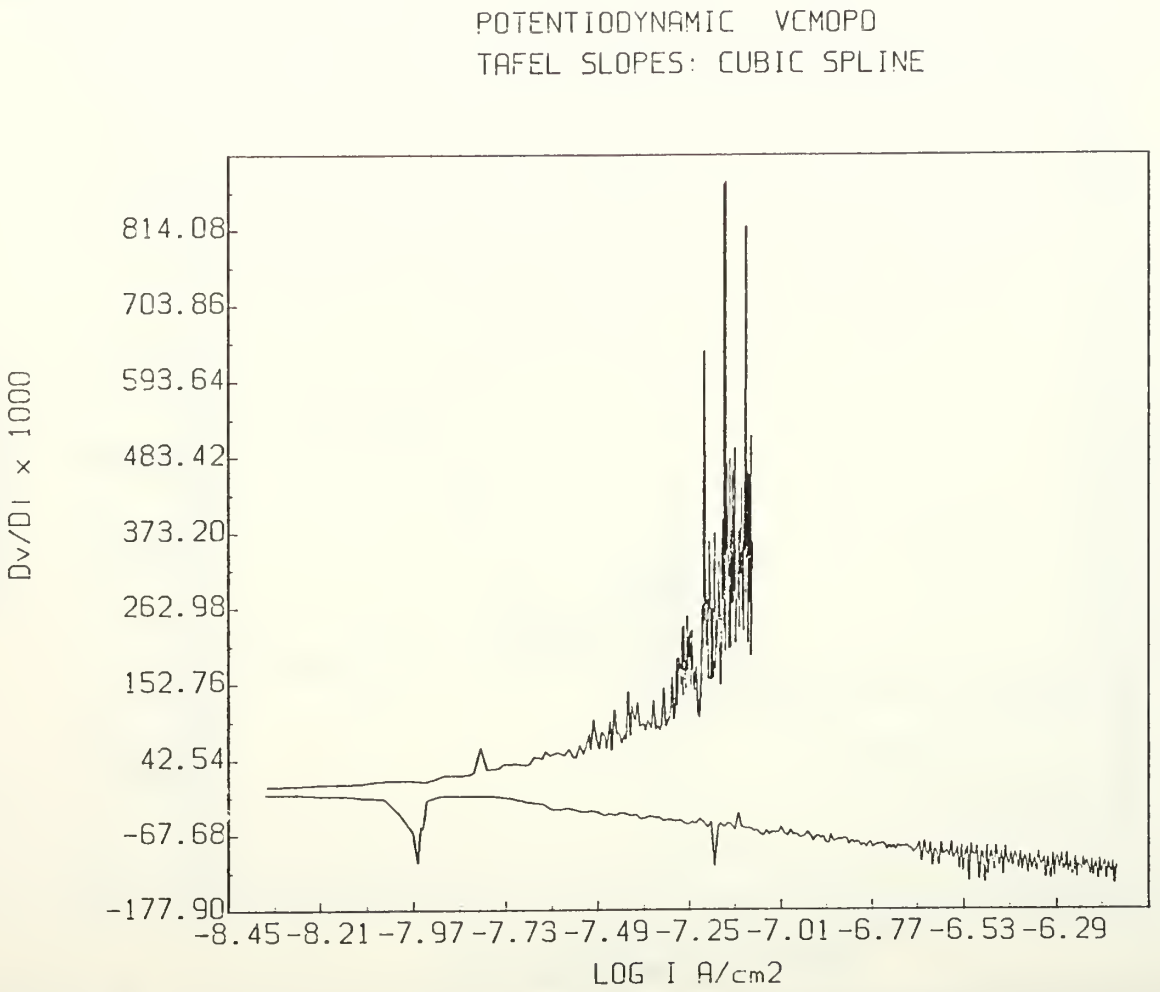

Figure 30. Cubic Spline derivatives of Fe-Cr-Mo alloy in 3.5% NaCl solution.

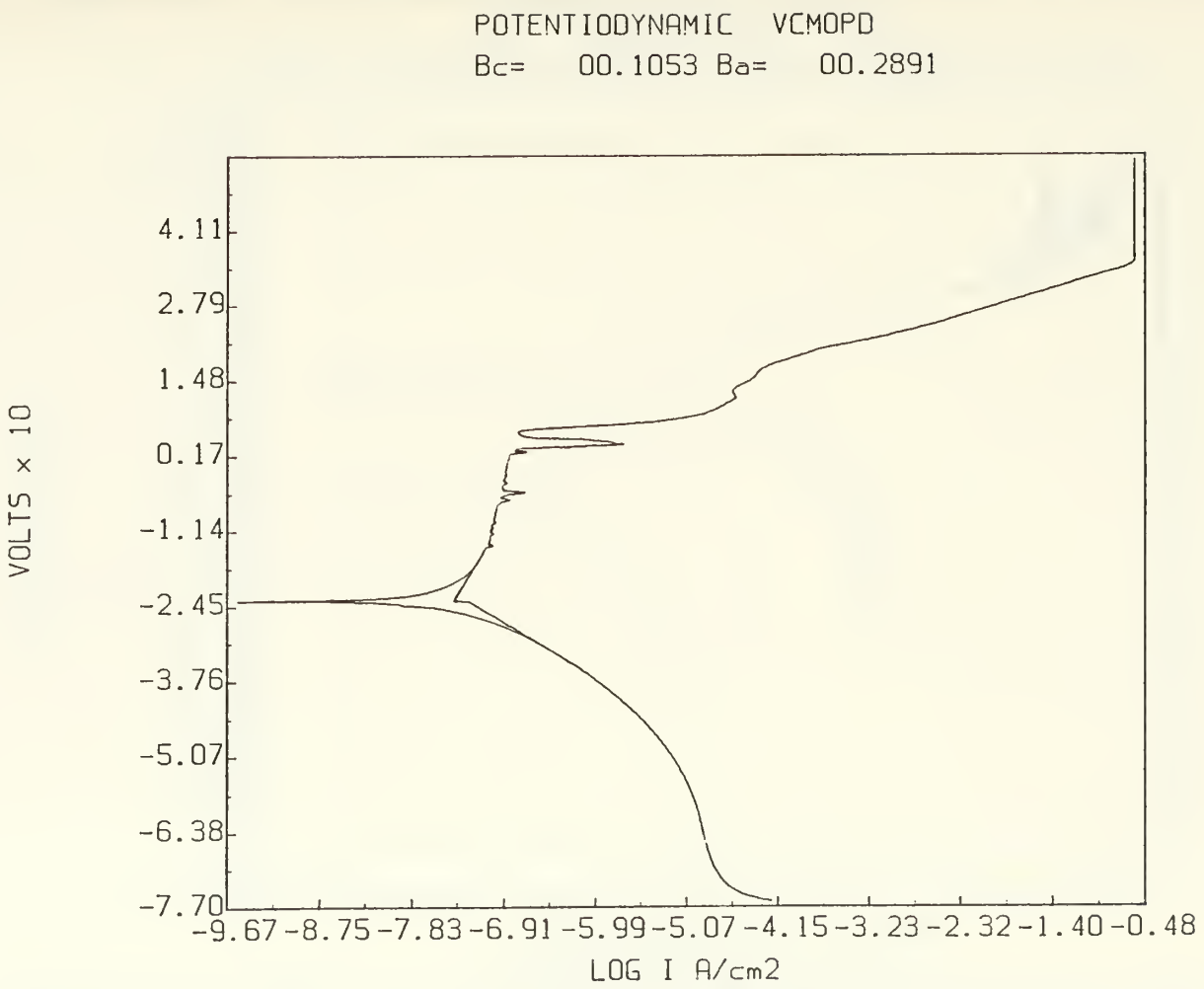

Figure 31. Tafel lines overlaid on PDP plot of Fe-Cr-Mo alloy in 3.5% NaCl solution.

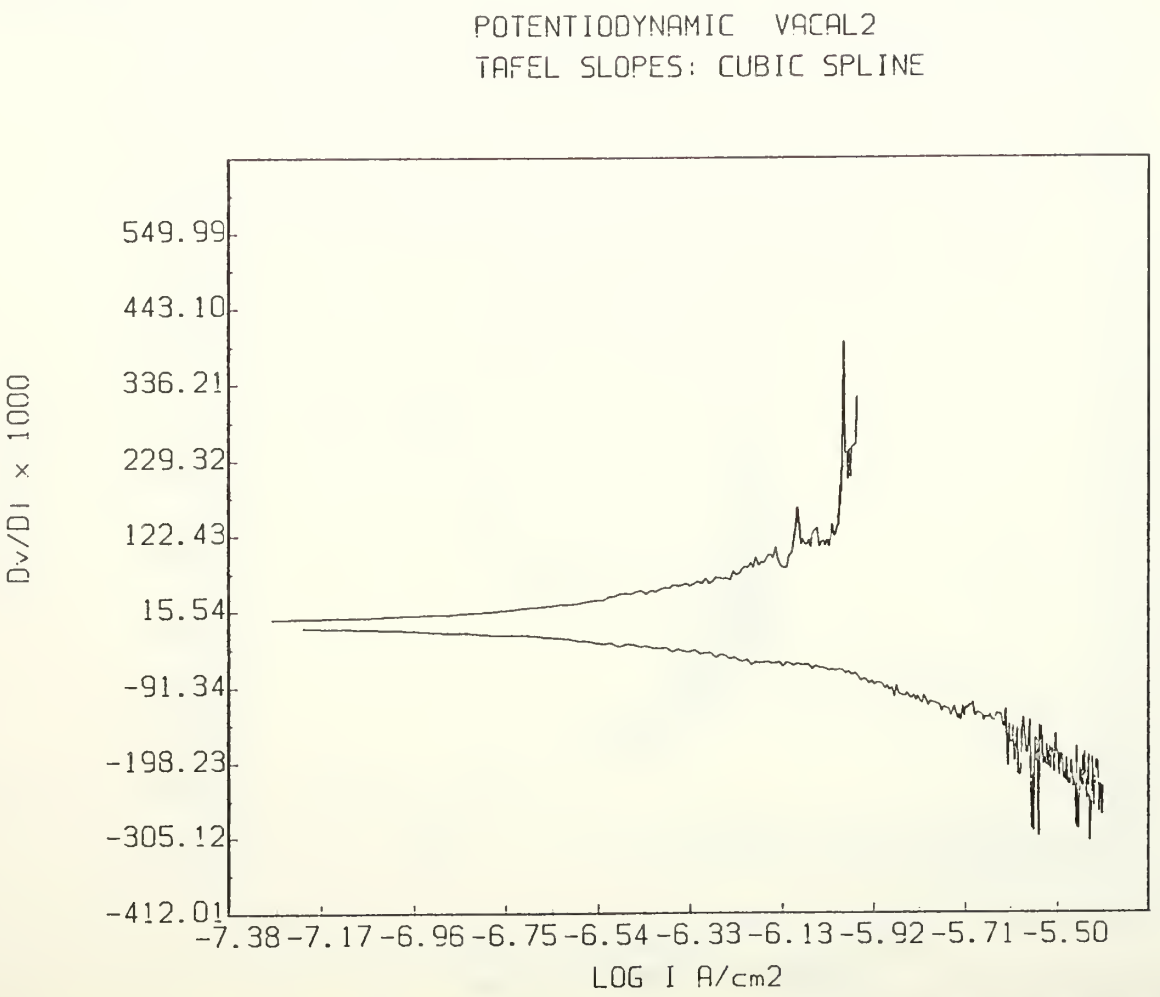

Figure 32. Cubic Spline derivatives of Fe-Cr-Al alloy in 3.5% NaCl solution.

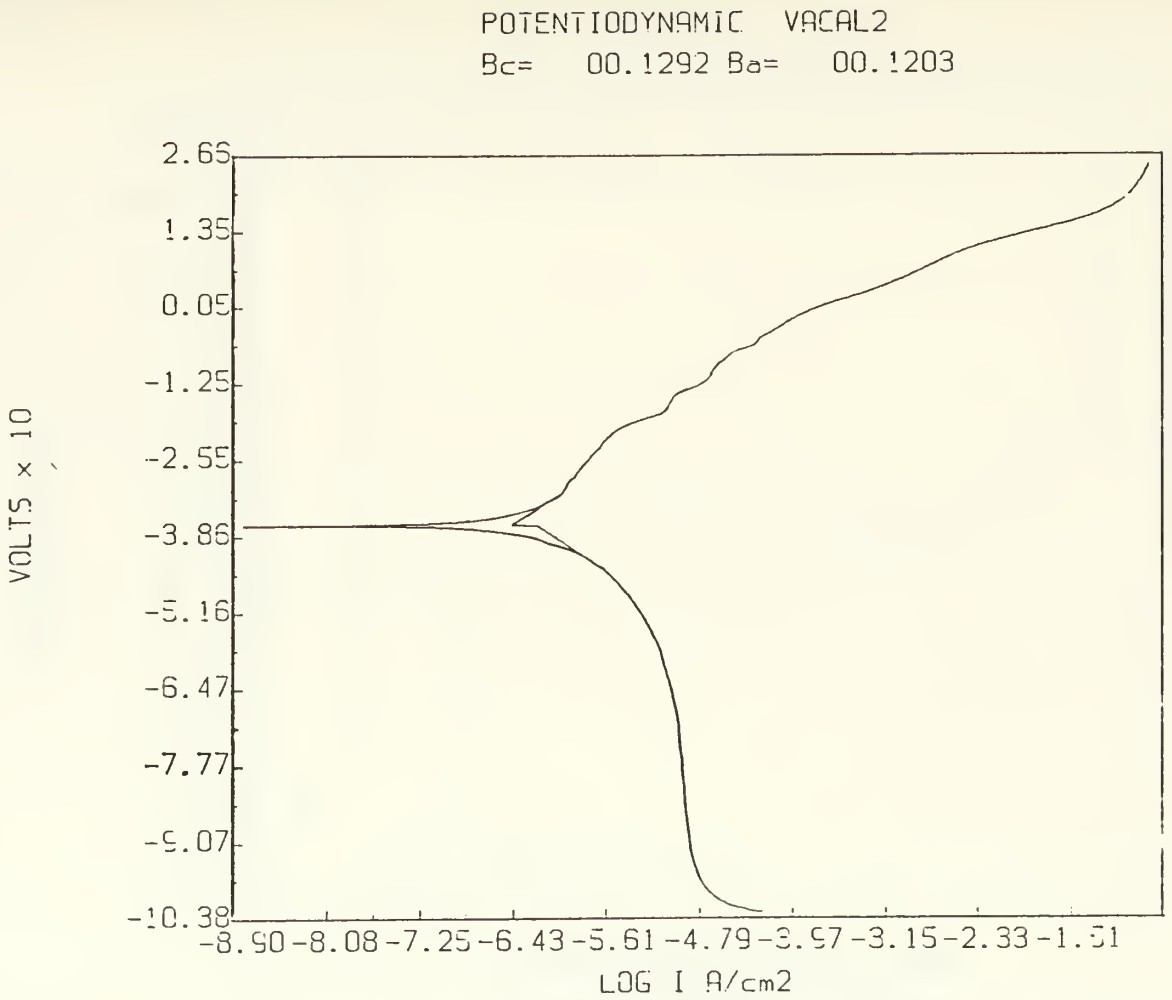

Figure 33. Tafel lines overlaid on PDP plot of Fe-Cr-Al alloy in 3.5% NaCl solution.

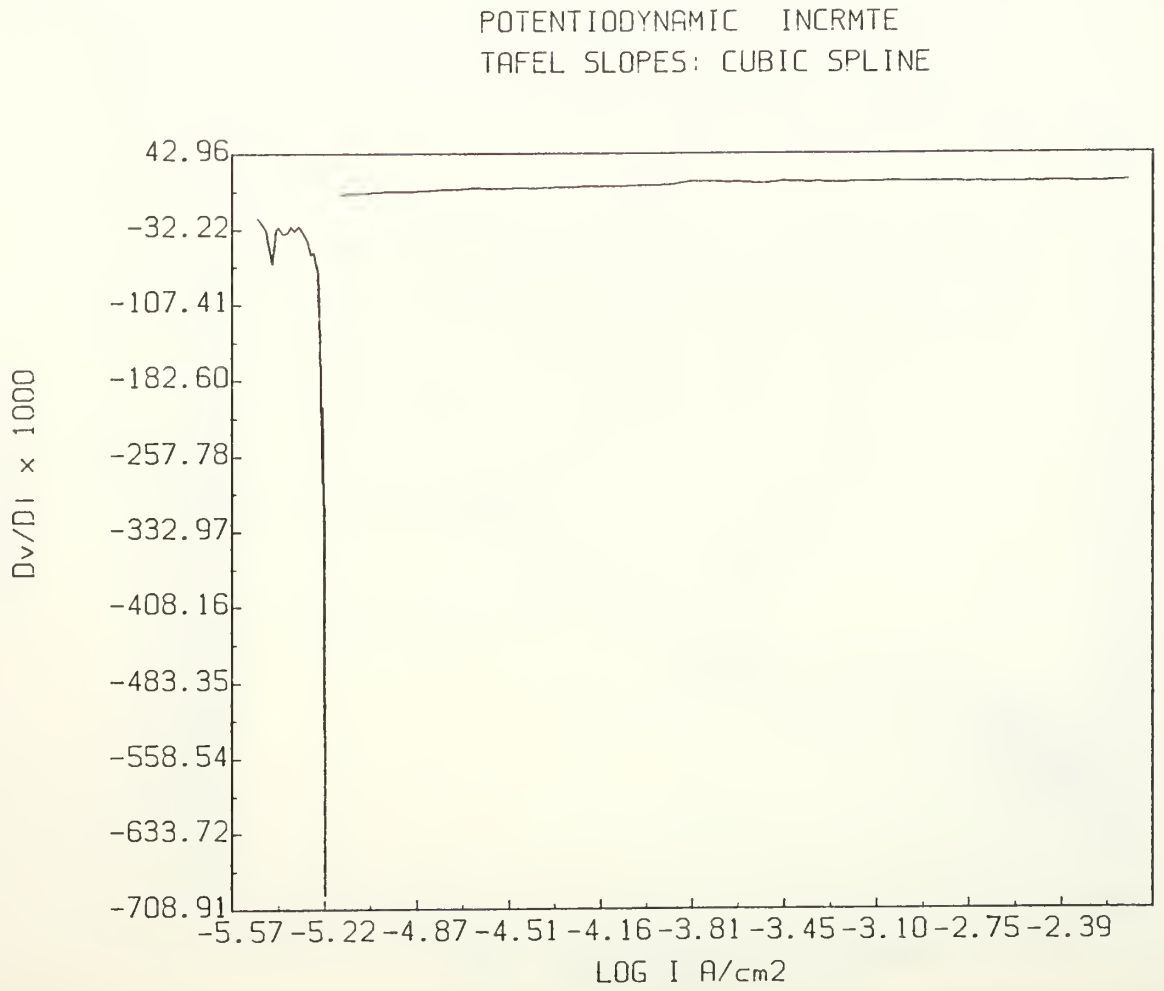

Figure 34. Cubic Spline derivatives of Cu-Mn-Al alloy in 3.5% NaCl solution.

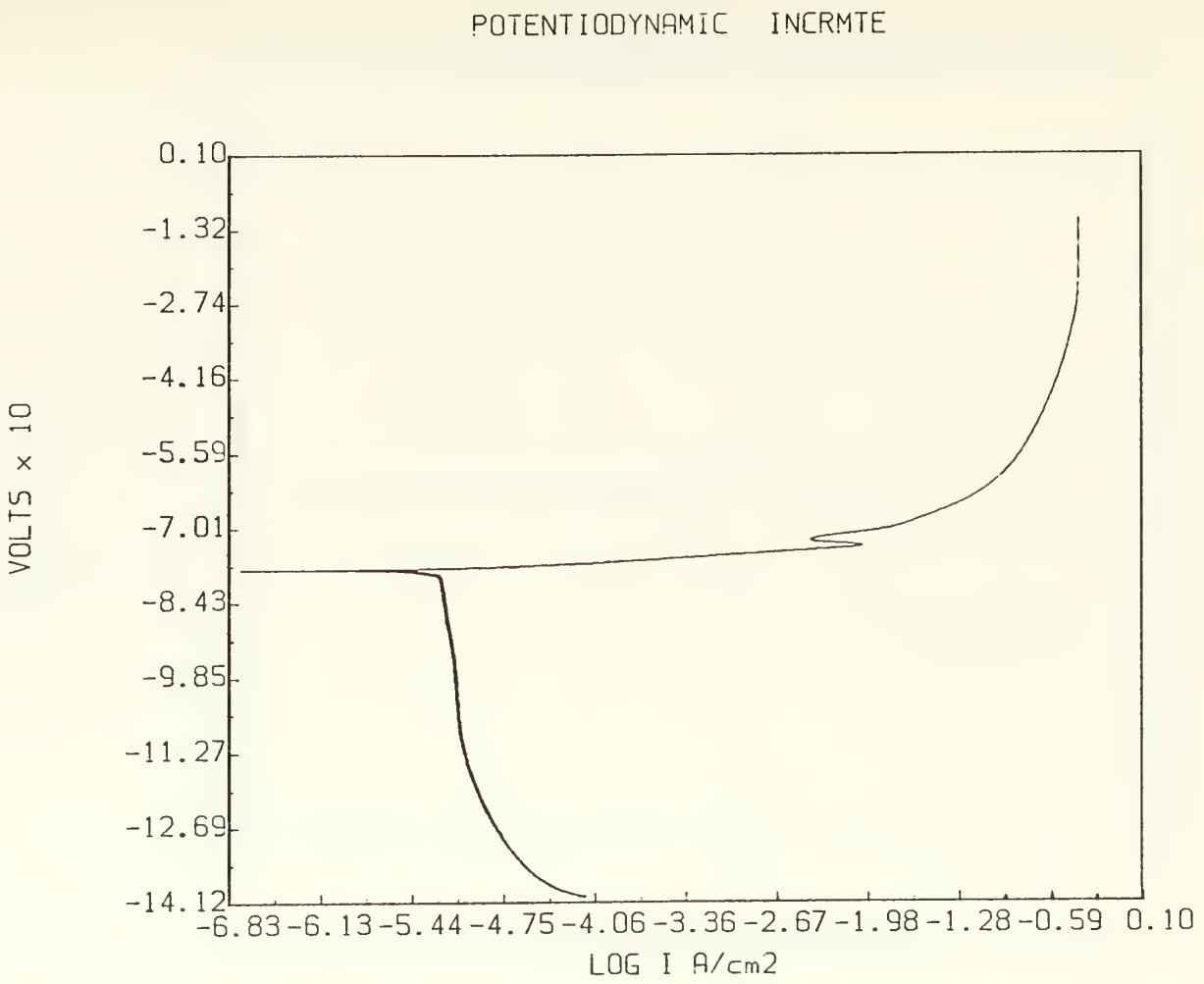

Figure 35. PDP plot of Cu-Mn-Al alloy in 3.5% NaCl solution.

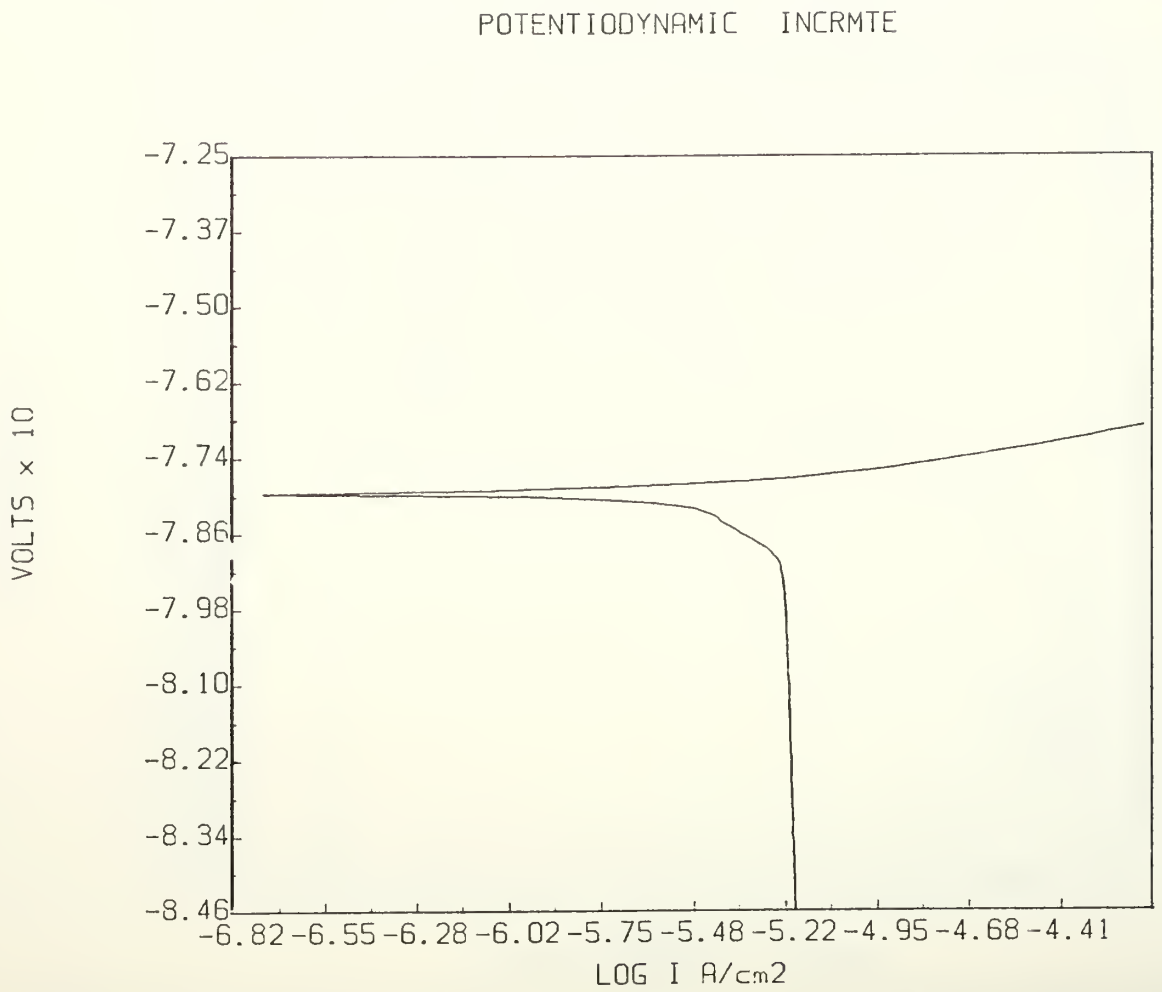

Figure 36. Concentration polarization of Cu-Mn-Al alloy in 3.5% NaCl solution.

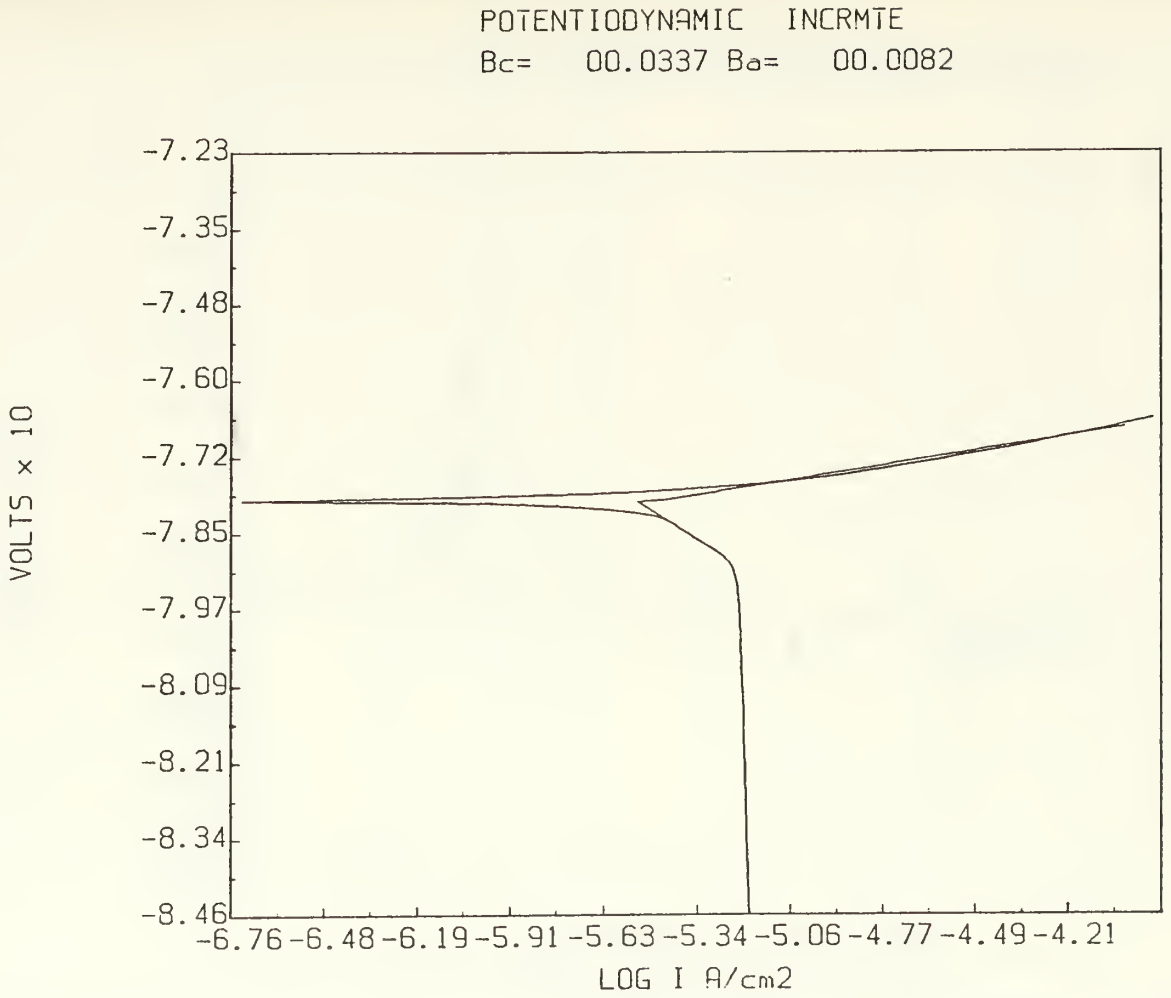

Figure 37. Tafel lines overlaid on PDP plot of Cu-Mn-Al alloy in 3.5% NaCl solution.

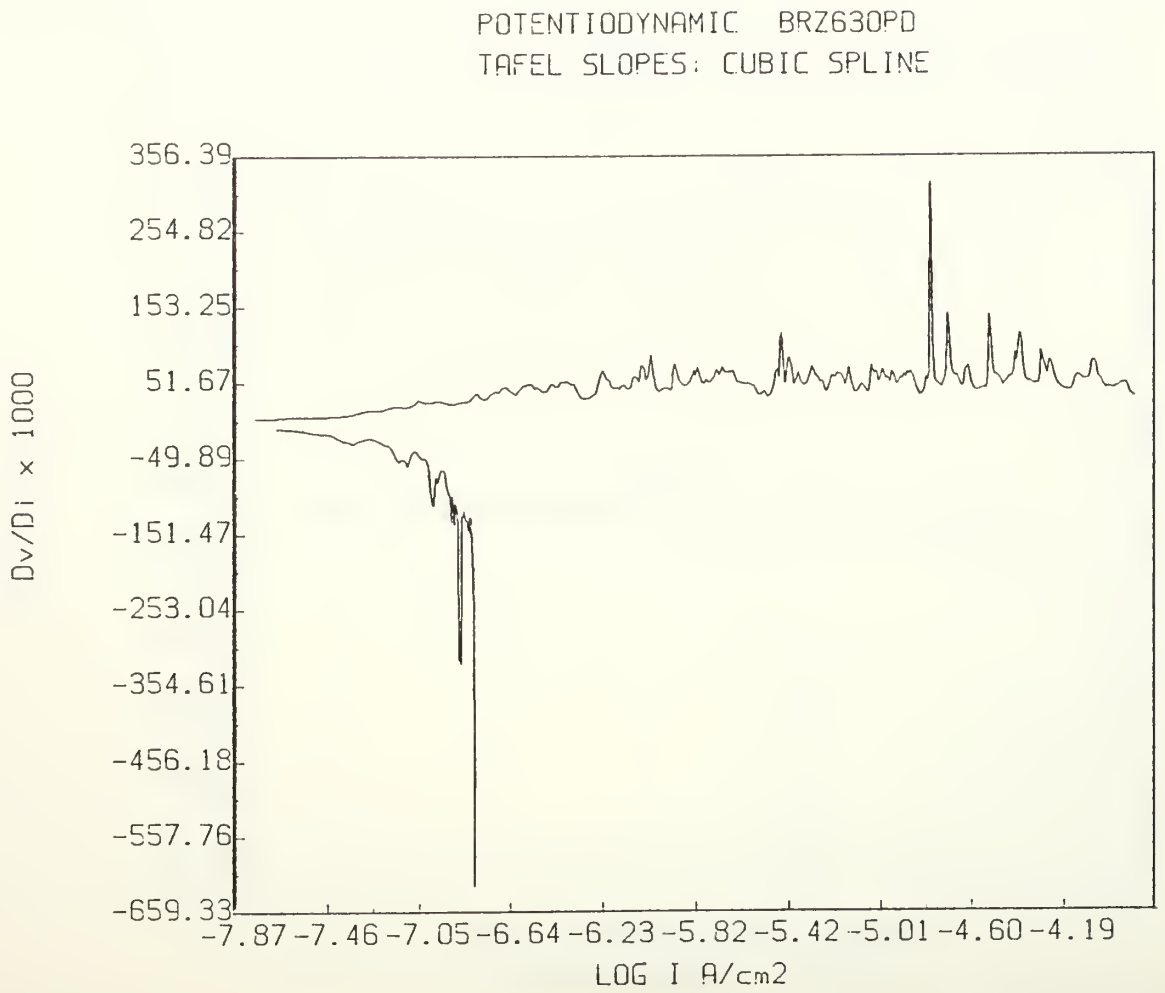

Figure 38. Cubic Spline derivatives of 630 series Bronze in 3.5% NaCl solution.

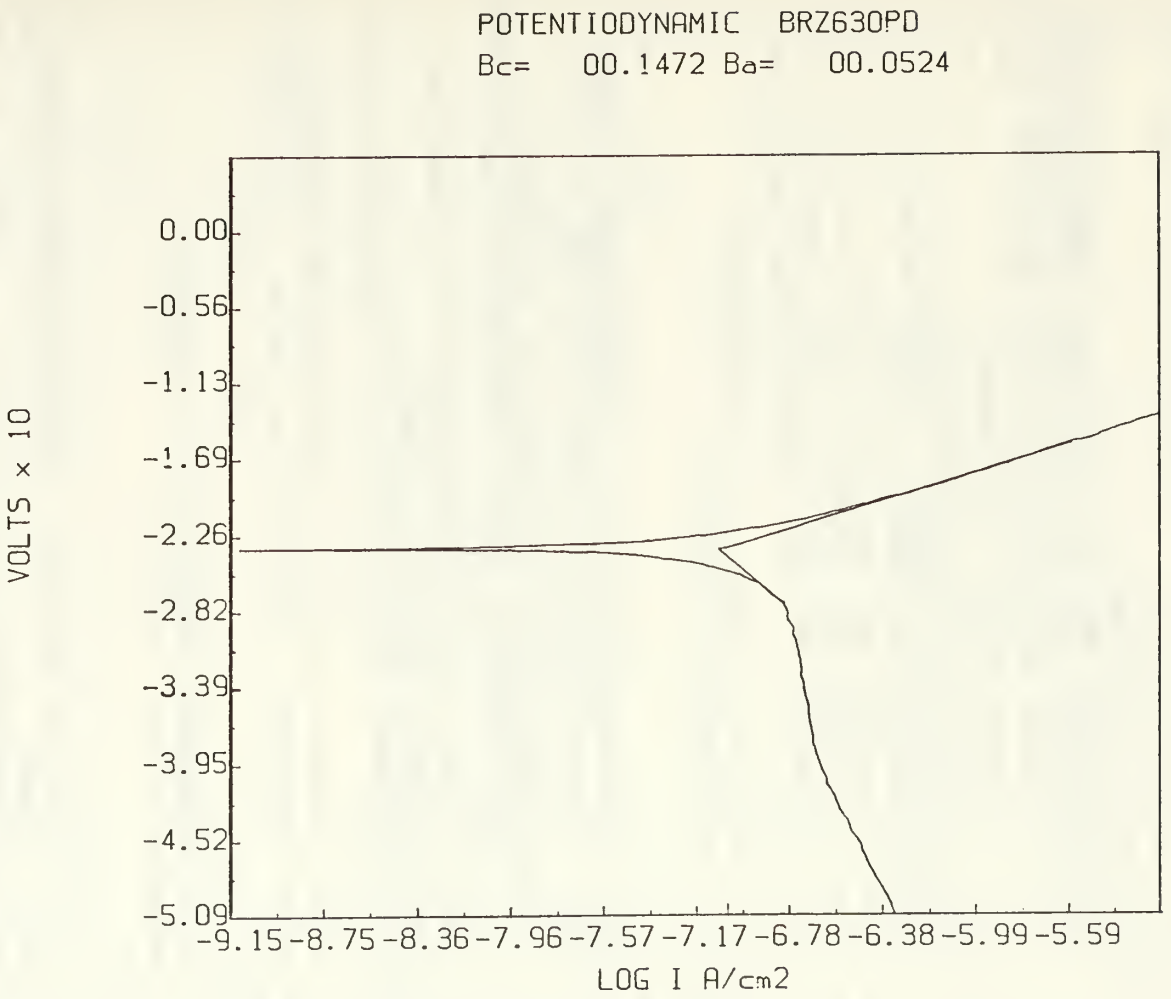

Figure 39. Tafel lines overlaid on PDP plot of 630 series Bronze in 3.5% NaCl solution.

APPENDIX B
PAR 351 TO Z-248 DATA TRANSFER PROGRAM

```
10 J = 0
20 SCREEN 0
30 COLOR 4,1,1
40 KEY OFF:CLS:CLOSE
50 DEFINT A-Z
60 LOCATE 25,1
70 PRINT STRING$(60," ")
80 FALSE = 0:TRUE=NOT FALSE
90 XOFF$=CHR$(19):XON$=CHR$(17)
100 LOCATE 25,20:PRINT "PAR 351 TO Z-248 DATA TRANSFER PROGRAM"
110 LOCATE 3,20:PRINT "YOU MUST PROPERLY CONFIGUR THE PAR 351."
120 LOCATE 5,20:PRINT "AFTER SETTING THE TIME AND DATE ON THE 351"
130 LOCATE 7,20:PRINT "PRESS 'SYSTEM MANAGER'"
140 LOCATE 8,20:PRINT "PRESS 'CONFIGUR SYSTEM'"
150 LOCATE 9,20:PRINT "SET 'PROTOCOL' TO PRINT"
160 LOCATE 10,20:PRINT "SET 'PARITY' TO NONE"
170 LOCATE 11,20:PRINT "SET 'BAUD RATE' TO 2400"
180 LOCATE 12,20:PRINT "SET 'STOP BIT' TO 1"
190 LOCATE 13,20:PRINT "SET 'WORD LENGTH' TO 8"
200 COLOR 14,1,1
210 LOCATE 15,20:PRINT "PRESS ANY KEY TO CONTINUE"
220 GO$ = INKEY$:IF GO$= "" GOTO 220
230 CLS
240 COLOR 4,1,1
250 LOCATE 25,20:PRINT "PAR 351 TO Z-248 DATA TRANSFER PROGRAM"
260 LOCATE 3,20:PRINT "PRESS 'MAIN MENU' ON THE 351"
270 LOCATE 4,20:PRINT "PRESS 'RECALL/DISPLAY EXPERIMENT'"
280 LOCATE 5,20:PRINT"PRESS 'DISPLAY EXPERIMENT'ON THE 'COPY FROM DISK'
SECTION"
290 LOCATE 6,20:PRINT "SELECT THE EXPERIMENT TO BE TRANSFERED"
300 LOCATE 7,20:PRINT "PRESS 'PLOT/FORMAT'"
310 LOCATE 8,20:PRINT "ENSURE THE X AXIS IS IN THE LINEAR SCALE"
320 COLOR 2,1,1
330 LOCATE 9,20:PRINT "IF NOT IN A LINEAR DISPLAY THEN PRESS"
340 LOCATE 10,20:PRINT "'FORMAT DISPLAY,'LINEAR','REVIEW DISPLAY','PLOT
FORMAT'"
350 COLOR 14,1,1
360 LOCATE 14,20:PRINT "PRESS ANY KEY TO CONTINUE"
370 GO$=INKEY$:IF GO$= "" GOTO 370
380 CLS
390 COLOR 4,1,1
400 LOCATE 25,20:PRINT "PAR 351 TO Z-248 DATA TRANSFER PROGRAM"
410 SPEED$ = "2400"
420 COMFIL$ ="COM2:"+SPEED$+",N,8,1,CS,DS"
430 OPEN COMFIL$ AS #1
```

```
440 PRINT #1, XOFF$;
450 OPEN "SCRN:" FOR OUTPUT AS #2
460 LOCATE 10,25:PRINT "(R)eceive a file OR (E)xit "
470 LOCATE 11,37:INPUT TXRX$
480 IF  (TXRX$<>"R") AND (TXRX$<>"E") THEN 460
490 IF TXRX$="E" THEN 1350
500 CLS
510 COLOR 4,1,1
520 LOCATE 25,20:PRINT "PAR 351 TO Z-248 DATA TRANSFER PROGRAM"
530 COLOR 15,1,1
540 LOCATE 13,25:PRINT "PRESS 'PRINT DATA' ON THE 351 THEN"
550 LOCATE 15,30:PRINT "PRESS ANY KEY TO CONTINUE"
560 GO$ = INKEY$:IF GO$="" GOTO 560
570 CLS
580 COLOR 4,1,1
590 LOCATE 25,20:PRINT "PAR 351 TO Z-248 DATA TRANSFER PROGRAM"
600 LOCATE 10,20:PRINT "THIS PROGRAM USES A RAM DISK TO SPEED
EXECUTION."
610 COLOR 2,1,1
620 LOCATE 13,20:PRINT "DO NOT ENTER THE DISK DRIVE OR THE FILE TYPE"
630 LOCATE 14,20:PRINT "BUT ONLY THE NAME OF THE OUTPUT FILE"
640 COLOR 14,1,1
650 LOCATE 16,20:PRINT "ENTER THE OUTPUT FILE NAME"
660 LOCATE 18,25:INPUT FIL$
670 DSK$="D:":DSKFIL$=DSK$+FIL$
680 OPEN "D:TMP" FOR OUTPUT AS #3
690 N = 0
700 CLS
710 PRINT #1, XON$;
720 IF LOC(1)=0 THEN GOSUB 820
730 IF LOC(1)>128 THEN PAUSE=TRUE:PRINT #1,XOFF$
740 A$=INPUT$(1,#1)
750 IF (A$=CHR$(10)) OR (A$=CHR$(13)) THEN B$=CHR$(13):J=J+1
760 IF (A$<>CHR$(13)) AND (A$<>CHR$(10)) THEN PRINT #3,A$;
770 IF (J > 3) AND (B$=CHR$(13)) AND (A$=CHR$(13)) THEN PRINT
#3,CHR$(13)
780 IF (J > 3) AND (B$=CHR$(13)) AND (A$=CHR$(13)) THEN B$=" ":N=N+1
790 IF LOC(1)>0 THEN 730
800 IF PAUSE THEN PAUSE=FALSE:PRINT #1,XON$;
810 GOTO 720
820 FOR I=1 TO 5000
830 IF LOC(1)<>0 THEN I = 9999
840 NEXT I
850 IF I>9999 THEN RETURN
860 CLOSE #3:CLS
870 F$="D:"+A$
880 OPEN "D:TMP" FOR INPUT AS #4
890 OPEN DSKFIL$ FOR OUTPUT AS #3
900 FOR I = 1 TO N
910 X$ = INPUT$(44,#4)
920 Y$ = INPUT$(2,#4)
```

```
930 B$ = MID$(X$,18,9)
940 C$ = MID$(B$,1,1)
950 D$ = MID$(B$,2,1)
960 E$ = MID$(B$,3,1)
970 F$ = MID$(B$,4,1)
980 G$ = MID$(B$,5,1)
990 H$ = MID$(B$,6,1)
1000 I$ = MID$(B$,7,1)
1010 J$ = MID$(B$,8,1)
1020 K$ = MID$(B$,9,1)
1030 FX$ = "-0.000000"
1040 IF (E$ = "E") THEN MID$(X$,18,9) = FX$
1050 IF (F$ = "E") THEN MID$(X$,18,9) = FX$
1060 IF (G$="E") THEN MID$(X$,18,9) = FX$
1070 IF (H$="E") THEN MID$(X$,18,9) = FX$
1080 IF (H$="E") THEN MID$(X$,18,9) = FX$
1090 IF (I$="E") THEN MID$(X$,18,9)=FX$
1100 IF (J$="E") THEN MID$(X$,18,9)=FX$
1110 IF (K$="E") THEN MID$(X$,18,9)=FX$
1120 F2$ = "E-12 "
1130 A$ = MID$(X$,42,1)
1140 A1$=MID$(X$,43,1)
1150 A2$=MID$(X$,34,1)
1160 A3$ = A$+A1$
1170 IF (A3$ = "12") AND (A2$ = "-") THEN MID$(X$,39,5) = F2$
1180 P$ = MID$(X$,1,1)
1190 IF (P$ = "P") THEN PRINT#3,X$
1200 NEXT I
1210 CLOSE #3:CLS
1220 J = 0
1230 COLOR 1,5,4
1240 BEEP
1250 LOCATE 12,30:PRINT "****TRANSFER COMPLETE****"
1260 LOCATE 13,30:PRINT "PRESS ANY KEY TO CONTINUE"
1270 GO$=INKEY$:IF GO$="" GOTO 1270
1280 PRINT #1, XOFF$
1290 CLOSE #1
1300 CLOSE #2
1310 CLOSE #3
1320 CLOSE #4
1330 COLOR 1,1,1
1340 GOTO 380
1350 END
```

APPENDIX C
CORROSION PROGRAM

```
*     THIS IS THE MAIN PROGRAM.  WITHIN IT THE USER ENTERS THE FILE
*     NAME WHICH HE/SHE WISHES TO ANALYZE.  FOLLOWING THIS A MENU
*     APPEARS ALLOWING THE USER TO SELECT THE DESIRED OPTION.
*
*     THE FOLLOWING VARIBLES ARE DEFINED
*
*     CDATA : THE 2 DIMENSIONAL ARRAY FOR THE DATA FILE CONTAINING
*             POTENTIAL AND CURRENT DENSITY
*     CDATA1: AS ABOVE BUT ONLY USED WHEN OVERLAYING TWO CURVES
*     MDATA : A TWO DIMENSIONAL ARRAY OF CDATA BUT NOW IN TERMS OF
*             POTENTIAL AND LOG10 CURRENT DENSITY
*     XDAT  : A ONE DIMENSIONAL ARRAY CONTAINING THE X VALUES
*             OF THE FIRST CURVE TO BE GRAPHED OR PLOTTED
*     YDAT  : A ONE DIMENSIONAL ARRAY CONTAINING THE Y VALUES
*             OF THE FIRST CURVE TO BE GRAPHED OR PLOTTED
*     XDAT1 : A ONE DIMENSIONAL ARRAY CONTAINING THE X VALUES
*             OF THE SECOND CURVE TO BE GRAPHED OR PLOTTED
*     YDAT1 : A ONE DIMENSIONAL ARRAY CONTAINING THE Y VALUES
*             OF THE SECOND CURVE TO BE GRAPHED OR PLOTTED
*     XI    : THE SUM OF THE LOG CURRENT DENSITIES. USED IN THE
*             LINEAR REGRESSION FOR DETERMINING THE TAFEL CONSTANTS
*     X2    : THE SUM OF THE SQUARES OF THE LOG CURRENT DENSITY
*             INITIALLY USED IN THE SAME MANNER A XI BUT ONCE THE
*             TAFEL CONSTANT HAS BEEN DETERMINED USED AS THE LOG
*             CURRENT DENSITY VALUE IN CREATING THE TAFEL LINES
*     XY    : THE SUM OF THE PRODUCTS OF LOG CURRENT DENSITY AND
*             POTENTIAL. USED IN THE SAME MANNER AS XI AND X2
*     Y1    : THE SUM OF THE POTENTIAL. USED AS XI,X2,XY
*     NS    : THE NUMBER OF POINTS USED TO GENERATE THE TAFEL SLOPE
*     B2    : THE TAFEL SLOPE AS CALCULATED IN THE REGRESSION
*     B1    : THE INTERCEPT OF THE TAFEL LINE
*     M     : MENU RESPONSE
*     I     : DO LOOP COUNTER
*     N     : THE NUMBER OF POINTS IN THE DATA FILE BEING READ
*     O1    : THE NUMBER OF POINTS IN THE FIRST DATA FILE, UNCHANGING
*     N1    : THE NUMBER OF POINTS IN THE FIRST DATA FILE. USED IN
*             THE SUBROUTINE SLOPE1 AND CAN BE RETURNED WITH A
*             DIFFERENT VALUE
*     J     : THE LOCATION OF ECORR WITHIN MDATA
*     J1    : SAME AS J BUT PASSED AS AN ARGUEMENT TO SLOPE1
*     O2    : SAME AS O1 BUT USED ONLY WHEN OVERLAYING DATA FILES
*     J2    : SAME AS J BUT USED ONLY WHEN OVERLAYING DATA FILES
*     S1    : THE POSITION WITHIN MDATA WHERE THE REQUIREMENTS OF
*             THE CENTRAL DIFFERNCE METHOD FAIL. CATHODIC BRANCH
```

```
*     S2    : SAME AS S1 BUT FOR THE ANODIC BRANCH
*     S3    : THE TOTAL NUMBER OF POINTS IN THE CATHODIC BRANCH
*             WHICH MEET THE REQUIREMENTS FOR USING THE CENTRAL
*             DIFFERENCE METHOD
*     S4    : SAME AS S3 BUT FOR THE ANODIC BRANCH
*     A1    : THE POSITION WITHIN XDAT1 AND YDAT1 WHICH MARKS THE
*           : BEGINNING OF THE TAFEL REGION
*     A2    : SAME AS A1 BUT THE END OF THE REGION
*     A3    : SERVES THE SAME PURPOSE AS A1 BUT WITH RESPECT TO
*             MDATA
*     A4    : SAME AS A3 BUT AS A2
*     C1    : SAME AS A1 BUT WITHIN XDAT AND YDAT (CATHODIC)
*     C2    : SAME AS C1 BUT THE END OF THE REGION
*     C3    : SERVES THE SAME PURPOSE AS C1 BUT WITH RESPECT TO
*             MDATA
*     C4    : SAME AS C3 BUT AS C2
*     N8    : SET TO 0. PASSED TO GRAPH1 AND PLOT TO AVOID THE USE
*             OF THE EMPTY XDAT1 AND YDAT1 ARRAYS WHEN THE ARE NOT
*             USED
*     NAME  : THE NAME OF THE MAIN DATA FILE IN USE
*     NAME2 : THE NAME OF THE SECOND FILE WHEN OVERLAYING
*     TITLE2: THE SECOND HEADING FOR THE GRAPH AND PLOT. CHANGES
*             DEPENDENT ON THE OPTION SELECTED
*     TITLE3: RESERVED FOR FUTURE USE IF DESIRED
*     D1    : CHARACTER STRING USED TO FORM TITLE2
*     D2    : THE CHARCACTER REPRESENTATION OF B2 WHEN B2 REPRESENT
*             THE CATHODIC TAFEL SLOPE
*     D3    : CHARACTER STRING USED TO FORM TITLE2
*     D4    : SAME AS D2 BUT FOR THE ANODIC TAFEL SLOPE
*     LIST  : CHARACTER STRING USED FOR MESSAGES AND PROMPTS


      REAL*4 XDAT(2000),YDAT(2000),XDAT1(2000),
     +YDAT1(2000)
      REAL*8 MDATA(2000,2),MDATA1(2000,2),X1,X2,N5,XY,XI,Y1
     +,CDATA(2000,2),CDATA1(2000,2)
      INTEGER*2 I,J,N,A1,A2,C1,C2,O1,NI,J1,S1,S2,S3,S4,O2,J2,N8
      INTEGER*2 A3,A4,C3,C4
      CHARACTER*8 NAME,NAME2, TITLE2*32,YNAME*5,LIST*50,TITLE3*50
      CHARACTER*8 D2,D4,D1*5,D3*6
      TITLE2 = ' '
      YNAME = ' '
C
C     THE SCREEN IS CLEARED AND THE FILE NAME IS ENTERED. THE USER
C     CAN ALSO EXIT THE PROGRAM OR GET A LIST OF EXISTING DATA FILES.
C     THIS PROMPT APPEARS IN PURPLE WHICH IS THE COLOR USED WHENEVER
C     A CHARACTER RESPONSE IS EXPECTED TO A PROMPT. IF A FILE NAME IS
C     ENTERED WHICH DOES NOT EXIST AN ERROR MESSAGE WILL APPEAR IN RED.
C
10    CALL QSMODE(3)
      CALL QCLEAR(0,5)
```

```
15    LIST = ' '
      CALL GCMOV(15,12)
      WRITE(6,805) LIST
      CALL GCMOV(25,12)
      LIST = 'ENTER THE FILE NAME TO BE USED'
      WRITE(6,805) LIST
      CALL GCMOV(20,11)
      LIST = 'A BLANK TO EXIT, OR "DIR" FOR A DIRECTORY'
      WRITE(6,805) LIST
      CALL GCMOV(36,10)
      READ(6,800) NAME
      CALL GCMOV(40,9)
      IF (NAME .EQ. '        ') THEN
         GOTO 900
      ENDIF
      IF (NAME .EQ. 'DIR') THEN
        CALL GCLEAR(0,5)
        CALL GCMOV(1,23)
        PAUSE 'ENTER "DIR *. /W" OR A BLANK TO CONTINUE'
        CALL GUSCRL(23,6,1,1,79,0,5)
        GOTO 15
      ENDIF
      CALL DATAIN(CDATA,N,NAME,IER)
      IF (IER .EQ. 1) THEN
         GOTO 10
      ENDIF
      CALL DATAM(CDATA,MDATA,N,J)
      CALL CHECK(MDATA,N,J,S1,S2,S3,S4)
20    CALL GSMODE(3)
      CALL GCLEAR(0,2)
      J1 = J
      N = N-1
      O1 = N
      N1 = N
C
C     THE MAIN MENU IS DISPLAYED IN GREEN.  THE COLOR GREEN IS USED
C     WHENEVER AN INTEGER IS EXPECTED AS THE RESPONSE TO A PROMPT.
C     IF THE INPUT IS NOT AN INTEGER OR A CHARACTER AN ERROR MESSAGE
C     IS DISPLAYED IN RED.
C
21    CALL GCMOV(15,22)
      LIST = ' '
      WRITE(6,805) LIST
      CALL GCMOV(15,18)
      LIST =  '1. PLOT ORIGINAL POTENTIODYNAMIC CURVE'
      WRITE(6,805) LIST
      CALL GCMOV(15,17)
      LIST =  '2. GENERATE TAFEL SLOPES USING CUBIC SPLINE METHOD'
      WRITE(6,805) LIST
      CALL GCMOV(15,16)
      LIST =  '3. GENERATE TAFEL SLOPES USING CENTRAL DIFFERENCE METHOD'
```

```
      WRITE(6,805) LIST
      CALL QCMOV(15,15)
      LIST =  '4. USE ANOTHER DATA FILE'
      WRITE(6,805) LIST
      LIST =  '5. OVERLAY TWO POTENTIODYNAMIC CURVES'
      CALL QCMOV(15,14)
      WRITE(6,805) LIST
      CALL QCMOV(15,13)
      LIST =  '6. EXIT'
      WRITE(6,805) LIST
      CALL QCMOV(29,12)
      LIST =  'ENTER 1,2,3,4,5 OR 6'
      WRITE(6,805) LIST
      CALL QCMOV(39,10)
      READ(6,810,ERR = 22,IOSTAT = J4) M
      CALL QCMOV(39,9)
22    IF ((J4 .NE. 0) .OR. (M .GT. 6)) THEN
         CALL QSMODE(3)
         CALL QCLEAR(0,4)
         LIST = ' '
         CALL QCMOV(20,17)
         WRITE(6,805) LIST
         LIST = 'YOU DID NOT ENTER AN INTEGER OF'
         CALL QCMOV(20,16)
         WRITE(6,805) LIST
         LIST = 'VALUE 1,2,3,4 OR 5.'
         CALL QCMOV(20,15)
         WRITE(6,805) LIST
         LIST = 'PRESS ANY KEY TO CONTINUE'
         CALL QCMOV(20,13)
         WRITE(6,805) LIST
         CALL QCMOV(39,12)
         CALL QINKEY(I3,I4)
         GOTO 20
      ENDIF
C
C     OPTION 1 HAS BEEN SELECTED. THIS GENERATES A GRAPH AND OPTIONAL
C     PLOT OF THE ORIGINAL POTENTIODYNAMIC CURVE.
C
        IF (M .EQ. 1) THEN
          DO 25 I = 1,01
            XDAT(I) = MDATA(I,2)
            YDAT(I) = MDATA(I,1)
25        CONTINUE
          N1 = N
          N8 = 0
          YNAME = 'VOLTS'
          TITLE2 = ' '
          TITLE3=' '
          CALL GRAPH1(XDAT,YDAT,01,J1,NAME,XDAT1,YDAT1,N8,YNAME,TITLE2
     +     ,A1,A2,C1,C2,TITLE3)
```

```
          GOTO 20
        ENDIF
C
C     OPTION 2 HAS BEEN SELECTED. THIS GENERATES THE DERIVATIVES OF
C     THE POTENTIODYNAMIC CURVE USING THE CUBIC SPLINE METHOD.
C
        IF (M .EQ. 2) THEN
          A1 = 0
          A2 = 0
          C1 = 0
          C2 = 0
30        J1 = J
          N1 = N
          CALL SLOPE1(MDATA,N1,J1,NAME,A1,A2,C1,C2)
C
C     STILL WITHIN OPTION 2. THE PROGRAM CHECKS TO ENSURE THAT THE
C     TAFEL REGIONS FOR BOTH THE ANODIC AND CATHODIC REGIONS HAVE
C     BEEN SELECTED.
C
          IF ((C1 .GE. C2) .OR. (A1 .GE. A2)) THEN
           CALL GSMODE(3)
           CALL GCLEAR(0,4)
           LIST = ' '
           CALL GCMOV(15,20)
           WRITE(6,805) LIST
           LIST = 'YOU DID NOT SELECT BOTH AN ANODIC'
           CALL GCMOV(15,19)
           WRITE(6,805) LIST
           LIST = 'AND CATHODIC BRANCH OR ELSE THE POINTS'
           CALL GCMOV(15,18)
           WRITE(6,805) LIST
           LIST = 'WERE NOT PROPERLY TRAPPED. IN ORDER TO'
           CALL GCMOV(15,17)
           WRITE(6,805) LIST
           LIST = 'PREVENT THE PROGRAM FROM CRASHING YOU'
           CALL GCMOV(15,16)
           WRITE(6,805) LIST
           LIST = 'MUST START AGAIN.'
           CALL GCMOV(15,15)
           WRITE(6,805) LIST
           LIST = 'PRESS ANY KEY TO CONTINUE'
           CALL GCMOV(15,13)
           WRITE(6,805) LIST
           CALL GINKEY(I3,I4)
           GOTO 20
          ENDIF
C
C     ALL REQUIRMENTS HAVE BEEN SATISFIED.  LINEAR REGRESSION IS
C     PERFORMED ON THE TAFEL REGIONS AND THE RESULTS ARE GRAPHED
C
          D1 = 'Bc= '
```

```
      YNAME = 'VOLTS'
      DO 35 I = 1,01
         XDAT(I) = MDATA(I,2)
         YDAT(I) = MDATA(I,1)
35    CONTINUE
      X2 = 0
      XI = 0
      XY = 0
      NI = ABS(C2-C1) +1
      YI = 0
      C3 = J-(4+C1)
      C4 = J-(4+C2)
      DO 40 I = C4,C3
        XI = XI + MDATA(I,2)
        X2 = X2 + (MDATA(I,2)**2)
        XY = XY + (MDATA(I,2)*MDATA(I,1))
        YI = YI+ MDATA(I,1)
40    CONTINUE
      NS = FLOAT(NI)
      B1 = ((XY*XI)-(YI*X2))/((XI*XI)-(NS*X2))

      B2 = ((XY*NS)-(YI*XI))/((X2*NS)-(XI*XI))
      Y1 = MDATA(J,1)
      X1 = ((Y1-B1)/B2)
      X2 = MDATA(C4,2)
      DX = (X2-X1)/(50.)
      DO 45 I = 1,50
        YDAT1(I) = (B2*X2)+B1
        XDAT1(I) = X2
        X2 = X2- DX
45    CONTINUE
      B2 = -B2
      CALL CONV(D2,B2)
      X2 = 0
      XI = 0
      XY = 0
      NS = 0
      YI = 0
      A3 = J+4+A1
      A4 = J+4+A2
      DO 50 I = A3,A4
        XI = XI + MDATA(I,2)
        X2 = X2 + (MDATA(I,2)**2)
        XY = XY + (MDATA(I,2)*MDATA(I,1))
        YI = YI+ MDATA(I,1)
50    CONTINUE
      NI = ABS(A3-A4)+1
      NS = FLOAT(NI)
      B1 = ((XY*XI)-(YI*X2))/((XI*XI)-(NS*X2))
      B2 = ((XY*NS)-(YI*XI))/((X2*NS)-(XI*XI))
      Y1 = MDATA(J,1)
```

```
            X1 = ((Y1-B1)/B2)
            X2 = MDATA(A4,2)
            DX = (X2-X1)/(50.)
            DO 55 I = 51,100
              YDAT1(I) = (B2*X1)+B1
              XDAT1(I) = X1
              X1 = X1+ DX
55          CONTINUE
            D3 = ' Ba= '
            CALL CONV(D4,B2)
            X2 = (XDAT1(500)+XDAT1(501))/2
            TITLE2=D1//D2//D3//D4
            NB = 100
            CALL GRAPH1(XDAT,YDAT,O1,J1,NAME,XDAT1,YDAT1,NB,YNAME,TITLE2
     +     ,A1,A2,C1,C2,TITLE3)
C
C      THIS PART ALLOWS THE USER TO SELECT EITHER ANOTHER ANODIC OR
C      CATHODIC TAFEL REGION OR TO REVIEW THE GRAPH WITHOUT HAVING
C      TO DETERMINE BOTH THE TAFEL REGIONS AGAIN.
C
56          CALL GCLEAR(0,2)
            LIST = ' '
            CALL GCMOV(20,16)
            WRITE(6,805) LIST
            LIST = 'IF THE GRAPH DID NOT SHOW A GOOD CORRELATION'
            CALL GCMOV(20,15)
            WRITE(6,805) LIST
            LIST = 'YOU CAN REPEAT THE CALCULATIONS FOR THE'
            CALL GCMOV(20,14)
            WRITE(6,805) LIST
            LIST = 'BRANCH IN QUESTION'
            CALL GCMOV(20,13)
            WRITE(6,805) LIST
            LIST = '1. REPEAT CALCULATIONS'
            CALL GCMOV(20,11)
            WRITE(6,805) LIST
            LIST = '2. EXIT THIS PORTION'
            CALL GCMOV(20,9)
            WRITE(6,805) LIST
            LIST = 'ENTER A 1 OR A 2'
            CALL GCMOV(20,7)
            WRITE(6,805) LIST
            CALL GCMOV(25,6)
            READ(6,810,ERR=56) M
            IF (M .EQ. 1) THEN
               GOTO 30
            ENDIF
            IF (M .EQ. 2) THEN
               GOTO 20
            ELSE
               GOTO 56
```

```
          ENDIF
        ENDIF

C     THE CENTRAL DIFFERNCE METHOD IS TO BE USED. THE OPTION PROCEEDS
C     IN THE SAME MANNER AS OPTION 2

        IF (M .EQ. 3) THEN
          A1 = 0
          A2 = 0
          C1 = 0
          C2 = 0
57        CALL SLOPE(MDATA,01,J1,NAME,A1,A2,C1,C2,S1,S2,S3,S4)
          IF ((C1 .GE. C2) .OR. (A1 .GE. A2)) THEN
           CALL QSMODE(3)
           CALL QCLEAR(0,4)
           LIST = ' '
           CALL QCMOV(15,20)
           WRITE(6,805) LIST
           LIST = 'YOU DID NOT SELECT BOTH AN ANODIC'
           CALL QCMOV(15,19)
           WRITE(6,805) LIST
           LIST = 'AND CATHODIC BRANCH OR ELSE THE POINTS'
           CALL QCMOV(15,18)
           WRITE(6,805) LIST
           LIST = 'WERE NOT PROPERLY TRAPPED. IN ORDER TO'
           CALL QCMOV(15,17)
           WRITE(6,805) LIST
           LIST = 'PREVENT THE PROGRAM FROM CRASHING YOU'
           CALL QCMOV(15,16)
           WRITE(6,805) LIST
           LIST = 'MUST START AGAIN.'
           CALL QCMOV(15,15)
           WRITE(6,805) LIST
           LIST = 'PRESS ANY KEY TO CONTINUE'
           CALL QCMOV(15,13)
           WRITE(6,805) LIST
           CALL QINKEY(I3,I4)
           GOTO 20
          ENDIF
          D1='Ec= '
          YNAME = 'VOLTS'
          DO 60 I = 1,01
             XDAT(I) = MDATA(I,2)
             YDAT(I) = MDATA(I,1)
60        CONTINUE
          X2 = 0
          XI = 0
          XY = 0
          NI = ABS(C2-C1) +1
          YI = 0
          C3 = J-(2+C1)
```

```
      C4 = J-(2+C2)
      DO 70 I = C4,C3
        XI = XI + MDATA(I,2)
        X2 = X2 + (MDATA(I,2)**2)
        XY = XY + (MDATA(I,2)*MDATA(I,1))
        YI = YI+ MDATA(I,1)
70    CONTINUE
      NS = FLOAT(NI)
      B1 = ((XY*XI)-(YI*X2))/((XI*XI)-(NS*X2))
      B2 = ((XY*NS)-(YI*XI))/((X2*NS)-(XI*XI))
      Y1 = MDATA(J,1)
      X1 = ((Y1-B1)/B2)
      X2 = MDATA(C4,2)
      DX = (X2-X1)/(50.)
      DO 80 I = 1,50
        YDAT1(I) = (B2*X2)+B1
        XDAT1(I) = X2
        X2 = X2- DX
80    CONTINUE
      B2 = -B2
      CALL CONV(D2,B2)
      D3 =' Ba= '
      X2 = 0
      XI = 0
      XY = 0
      NS = 0
      YI = 0
      A3 = J+2+A1
      A4 = J+2+A2
      DO 90 I = A3,A4
        XI = XI + MDATA(I,2)
        X2 = X2 + (MDATA(I,2)**2)
        XY = XY + (MDATA(I,2)*MDATA(I,1))
        YI = YI+ MDATA(I,1)
90    CONTINUE
      NI = ABS(A2-A1)+1
      NS = FLOAT(NI)
      B1 = ((XY*XI)-(YI*X2))/((XI*XI)-(NS*X2))
      B2 = ((XY*NS)-(YI*XI))/((X2*NS)-(XI*XI))
      Y1 = MDATA(J,1)
      X1 = ((Y1-B1)/B2)
      X2 = MDATA(A4,2)
      DX = (X2-X1)/(50.)
      DO 100 I = 51,100
        YDAT1(I) = (B2*X1)+B1
        XDAT1(I) = X1
        X1 = X1+ DX
100   CONTINUE
      CALL CONV(D4,B2)
      TITLE2=D1//D2//D3//D4
      NB = 100
```

```
          CALL GRAPH1(XDAT,YDAT,O1,J1,NAME,XDAT1,YDAT1,N8,YNAME,TITLE2
     +    ,A1,A2,C1,C2,TITLE3)
58         CALL GCLEAR(0,2)
           LIST = ' '
           CALL GCMOV(20,16)
           WRITE(6,805) LIST
           LIST = 'IF THE GRAPH DID NOT SHOW A GOOD CORRELATION'
           CALL GCMOV(20,15)
           WRITE(6,805) LIST
           LIST = 'YOU CAN REPEAT THE CALCULATIONS FOR THE'
           CALL GCMOV(20,14)
           WRITE(6,805) LIST
           LIST = 'BRANCH IN QUESTION'
           CALL GCMOV(20,13)
           WRITE(6,805) LIST
           LIST = '1. REPEAT CALCULATIONS'
           CALL GCMOV(20,11)
           WRITE(6,805) LIST
           LIST = '2. EXIT THIS PORTION'
           CALL GCMOV(20,9)
           WRITE(6,805) LIST
           LIST = 'ENTER A 1 OR A 2'
           CALL GCMOV(20,7)
           WRITE(6,805) LIST
           CALL GCMOV(25,6)
           READ(6,810,ERR=58) M
           IF (M .EQ. 1) THEN
              GOTO 57
           ENDIF
           IF (M .EQ. 2) THEN
              GOTO 20
           ELSE
              GOTO 58
           ENDIF
         ENDIF

C      OPTION 4 RETURNS THE USER TO THE BEGINNING OF THE PROGRAM

         IF (M .EQ. 4) THEN
           GOTO 10
         ENDIF

C      OPTION 5 IS THE OVERLAY. ONLY THE POTENTIODYNAMIC CURVES CAN
C      BE GRAPHED OR PLOTTED. THE ORIGINAL FILE IS STILL AVAILABLE
C      FOR FURTHER MANIPULATION

         IF (M .EQ. 5) THEN
           TITLE3=' '
105        CALL GCLEAR(0,5)
           LIST = ' '
           CALL GCMOV(15,20)
```

```
      WRITE(6,805) LIST
      CALL GCMOV(25,12)
      LIST = 'ENTER THE FILE NAME TO BE USED'
      WRITE(6,805) LIST
      CALL GCMOV(20,11)
      LIST = 'A BLANK TO EXIT, OR "DIR" FOR A DIRECTORY'
      WRITE(6,805) LIST
      CALL GCMOV(36,10)
      READ(6,800) NAME2
      CALL GCMOV(40,9)
      IF (NAME2 .EQ. '        ') THEN
         GOTO 21
      ENDIF
      IF (NAME2 .EQ. 'DIR') THEN
        CALL GCLEAR(0,5)
        CALL GCMOV(1,23)
        PAUSE 'ENTER "DIR *. /W" OR A BLANK TO CONTINUE'
        CALL GUSCRL(23,6,1,1,79,0,5)
        GOTO 105
      ENDIF
      DO 750 I = 1,01
        XDAT(I) = MDATA(I,2)
        YDAT(I) = MDATA(I,1)
750   CONTINUE
      CALL DATAIN(CDATA1,02,NAME2,IER)
      IF (IER .EQ. 1) THEN
        GOTO 105
      ENDIF
      CALL DATAM(CDATA1,MDATA1,02,J2)
      CALL CHECK(MDATA1,02,J2,S1,S2,S3,S4)
      DO 760 I=1,02
        XDAT1(I) = MDATA1(I,2)
        YDAT1(I) = MDATA1(I,1)
760   CONTINUE
      TITLE2 ='OVERLAY '//NAME//' '//NAME2
      YNAME = 'VOLTS'
      NAME2=NAME
      NAME = ' '
      CALL GRAPH1(XDAT,YDAT,01,J1,NAME,XDAT1,YDAT1,02,YNAME,TITLE2,
     +  A1,A2,C1,C2,TITLE3)
        NAME = NAME2
       CALL GCLEAR(0,2)
       GOTO 21
    ENDIF
      IF (M .EQ. 6) THEN
        GOTO 900
      ELSE
        GOTO 20
      ENDIF
800   FORMAT(A8)
805   FORMAT(A50,\)
```

```
810     FORMAT(I1)
900     CALL GSMODE(3)
        END
```

```
*   THIS SUBROUTINE READS THE DATA FILE GENERTATED BY THE
*   MODEL 351 WHICH HAS BEEN TRANSFERRED UNTO A DISK

C       CDATA  = THE ARRAY WHICH CONTAINING POTENTIAL AND
C                CURRENT DENSITY
C       NAME   = THE NAME OF THE DATA FILE
C       C1     = CHARACTER STRING PRESENT IN THE DATA FILE
C       C2     = CHARACTER STRING PRESENT IN THE DATA FILE
C       N      = NUMBER OF DATA POINTS IN THE FILE,
C                DETERMINED DURING THE PROGRAM

        SUBROUTINE DATAIN (CDATA,N,NAME,IER)
        REAL*8 CDATA(2000,2)
        INTEGER*2, N,I
        CHARACTER C1*17,C2*6,NAME*8,LIST*50,C3*50
        OPEN (UNIT = 10, ERR = 200,FILE = NAME, STATUS = 'OLD')
        DO 50 N=1,2000
          READ (10,900,ERR=60) C1,CDATA(N,1),C2,CDATA(N,2)
50      CONTINUE
60      REWIND(10)
        DO 70 I = 1,2000
          READ(10,*,ERR = 100)
70      CONTINUE
100     IF (I .NE. N) THEN
           CALL GSMODE(3)
           CALL GCLEAR(0,4)
           LIST = ' '
           CALL GCMOV(15,20)
           WRITE(6,905) LIST
           LIST = 'AN ERROR HAS BEEN ENCOUNTERED IN'
           CALL GCMOV(15,19)
           WRITE(6,905) LIST
           LIST = 'THE FORMAT OF THE DATA FILE.'
           CALL GCMOV(15,18)
           WRITE(6,905) LIST
           LIST = 'THIS USUALLY INVOLVES A SWITCH BETWEEN'
           CALL GCMOV(15,17)
           WRITE(6,905) LIST
           LIST = 'FLOATING POINT AND EXPONENTIAL NOTATION'
           CALL GCMOV(15,16)
           WRITE(6,905) LIST
           LIST = 'AROUND ECOOR.'
           CALL GCMOV(15,15)
           WRITE(6,905) LIST
           LIST = 'TO CORRECT THIS PROBLEM EXIT THE PROGRAM'
           CALL GCMOV(15,13)
           WRITE(6,905) LIST
           LIST = 'AND USE THE DOS "EDLIN" FEATURE. THE'
```

```
            CALL QCMOV(15,12)
            WRITE(6,905) LIST
            LIST = 'PROBLEM BEGINS AT LINE #:'
            CALL QCMOV(15,11)
            WRITE(6,905) LIST
            CALL QCMOV(38,11)
            WRITE(6,915) N
            LIST = 'PRESS ANY KEY TO CONTINUE'
            CALL QCMOV(15,8)
            WRITE(6,905) LIST
            CALL QINKEY(I3,I4)
            GOTO 990
         ENDIF
         GOTO 990
200      CALL QSMODE(3)
         CALL QCLEAR(0,4)
         LIST = ' '
         CALL QCMOV(15,20)
         WRITE(6,905) LIST
         LIST = 'THE FILE NAME ENTERED DOES NOT EXIST.'
         CALL QCMOV(15,19)
         WRITE(6,905) LIST
         LIST = 'TO EXAMINE THE DIRECTORY OF POSSIBLE FILES'
         CALL QCMOV(15,18)
         WRITE(6,905) LIST
         LIST = 'ENTER "DIR" WHEN ASKED TO ENTER THE FILE'
         CALL QCMOV(15,17)
         WRITE(6,905) LIST
         LIST = 'NAME FOLLOWED BY THE DOS COMMAND.'
         CALL QCMOV(6,905)
         WRITE(6,905) LIST
         LIST = '"DIR *. /W"'
         CALL QCMOV(35,16)
         WRITE(6,905) LIST
         LIST = 'PRESS ANY KEY TO CONTINUE'
         CALL QCMOV(15,14)
         WRITE (6,905) LIST
         IER = 1
         CALL QINKEY(I3,I4)
         GOTO 990
900      FORMAT (A17,G8.4,A7,G10.4)
905      FORMAT(A50,\)
915      FORMAT(I4,\)
990      RETURN
         END
```

```
*  THIS SUBROUTINE MANIPULATES THE DATA IN THE ARRAY CDATA
*  THE PROPER FORM AND IDTENIFIES ECORR

C      CDATA  = THE ARRAY CONTAINING POTENTAIL AND CURRENT
C               DENSITY
C      MDATA  = CDATA  CONVERTED TO POTENTIAL AND LOG
C               CURRENT DENSITY
C      N      = NUMBER OF DATA POINTS IN THE DATA FILE
C      J      = LOCATION WITHIN MDATA WHERE ECORR IS LOCATED
C      I      = DO LOOP COUNTER
C      X1     = ABSOLUTE VALUE OF CURRENT DENSITY AT I
C               USED FOR DETERMINING ECORR
C      X2     = ABSOLUTE VALUE OF CURRENT DENSITY AT I-1
C               USED FOR DETERMINING ECORR
C      X3     = LOG10(X1)

       SUBROUTINE DATAM (CDATA,MDATA,N,J)
       REAL*8 MDATA(2000,2),CDATA(2000,2)
       INTEGER*2 N,J
       X2 = 100000.

C      CONVERT CURRENT DENSITY INTO LOG CURRENT DENSITY

       DO 100 I=1,N-2
          X1 = ABS(CDATA(I,2))
          X3 = LOG10(X1)

C      IF/THEN TEST USED IN DETERMINING LOCATION OF ECOOR

          IF ( X1 .LT. X2) THEN
             J = I
             X2 = X1
          ELSE
             X2 = X2
          ENDIF
          MDATA(I,1) = CDATA(I,1)
          MDATA(I,2) = X3
100    CONTINUE
       RETURN
       END
```

```
*     THIS SUBROUTINE PERFORMS TWO FUNCTIONS. ITS FIRST FUNCTION
*     IS TO CHECK TO ENSURE THAT THE POTENTIAL VALUES RECORDED
*     ACCURATELY REFLECT THE ACTUAL POLARIZATION. THE SECOND IS
*     TO DETERMINE THE NUMBER OF POINTS IN THE ANODIC AND CATHODIC
*     BRANCHES WHICH MAY BE USED IN THE CENTRAL DIFFERNCE METHOD.
*
*     MDATA : THAT ARRAY CONTAINING POTENTIAL AND LOG CURRENT
*             DENSITY
*     N3    : THE NUMBER OF POINTS IN THE CATHODIC BRANCH WHICH
*             CAN BE USED IN THE CENTRAL DIFFERENCE METHOD
*     N4    : THE NUMBER OF POINTS IN THE ANODIC BRANCH WHICH
*             CAN BE USED IN THE CENTRAL DIFFERNCE METHOD
*
*     OTHER VARIBLES ARE USED MERELY IN CHECKING REQUIRED CONDITIONS
*
      SUBROUTINE CHECK(MDATA,N,J,N1,N2,N3,N4)
      REAL*8 MDATA(2000,2)
      INTEGER*2 I,J,N,N1,N2,N3,N4
      CHARACTER*50 LIST

C     THE FIRST INCREMENT IN POTENTIAL FOR THE CATHODIC BRANCH  IS
C     DETERMINED

10    Y = MDATA(J,1)
      Y1 = MDATA(J-1,1)
      Z1 = ABS(ABS(Y1)-ABS(Y))
      Z1 = ANINT(Z1/0.0001)

C     SUBSEQUENT POTENTIAL INCREMENTS, WHICH MUST BE EQUAL TO THE
C     FIRST ARE CHECKED.  LOG CURRENT DENSITY VALUES ARE ALSO CHECKED
C     TO ENSURE THAT THEY WILL NOT RESULT IN THE CALCULATION OF A
C     SLOPE OF INFINITE VALUE.

      DO 20 I = J,2,-1
         Y = MDATA(I,1)
         Y1 = MDATA(I-1,1)
         X = MDATA(I,2)
         X1 = MDATA(I-1,2)
         Z = ABS(ABS(Y1)-ABS(Y))
         Z = ANINT(Z/0.0001)

C     ONE OF THE TWO REQUIRED CONDITIONS HAS FAILED.

           IF ((Z .NE. Z1) .OR. ( X1 .LE. X )) GOTO 25

20    CONTINUE
25    N1 = I+2
```

```
C     THE FIRST INCREMENT IN POTENTIAL FOR THE ANODIC BRANCH IS
C     DETERMINED

      X = MDATA(J+1,1)
      Y = MDATA(J+2,1)
      Z1 = ABS(X-Y)
      Z1 = ANINT(Z1/0.0001)

C     SUBSEQUENT POTENTIAL INCREMENTS, WHICH MUST BE EQUAL TO THE
C     FIRST ARE CHECKED.  LOG CURRENT DENSITY VALUES ARE ALSO CHECKED
C     TO ENSURE THAT THEY WILL NOT RESULT IN THE CALCULATION OF A
C     SLOPE OF INFINITE VALUE.

      DO 30 I = J+2,N-1
      Y = MDATA(I,1)
         Y1 = MDATA(I+1,1)
         X = MDATA(I,2)
         X1 = MDATA(I+1,2)
         Z = ABS(Y1-Y)
         Z = ANINT(Z/0.0001)

C     ONE OF THE TWO REQUIRED CONDITIONS HAS FAILED.

           IF ((Z .NE. Z1) .OR. ( X1 .LE. X )) GOTO 35

30    CONTINUE
35    N2 = I - 2
      N3 = J - N1
      N4 = N2 - J

C     THE NUMBER OF POINTS AVAILABLE FOR EACH BRANCH IS COMPARED
C     WITH A MINUMIUM NUMBER OF 15. IF NEITHER BRANCH HAS ENOUGH
C     AVAILABLE POINTS VARIOUS MESSAGES ARE DISPLAYED LISTING
C     POSSIBLE CAUSES AND OPTIONS.

      IF ((N3 .LE. 15) .OR. (N4 .LE. 15)) THEN
          CALL GCLEAR(0,4)
          LIST = ' '
          CALL GCMOV(20,20)
          WRITE(6,805) LIST
          LIST = 'THE CENTRAL DIFFERNCE METHOD REQUIRES THAT'
          CALL GCMOV(20,19)
          WRITE(6,805) LIST
          LIST = 'THE CURRENT BE MEASURED AT EQUAL DIFFERENTIALS'
          CALL GCMOV(20,18)
          WRITE(6,805) LIST
          LIST = 'OF THE APPLIED VOLTAGE. THE PAR MODEL 351'
          CALL GCMOV(20,17)
          WRITE(6,805) LIST
          LIST = 'RECORDS THE CURRENT AT INCREMENTS OF 2.0 mV'
          CALL GCMOV(20,16)
```

```
      WRITE(6,805) LIST
      LIST = 'UNTIL THE CHANGE IN CURRENT EXCEEDS A PRESCRIBED'
      CALL QCMOV(20,15)
      WRITE(6,805) LIST
      LIST = 'LEVEL. AT THAT POINT THE 351 BEGINS RECORDING'
      CALL QCMOV(20,14)
      WRITE(6,805) LIST
      LIST = 'THE CURRENT AT INCREMENTS OF 0.5 mV. AT AN'
      CALL QCMOV(20,13)
      WRITE(6,805) LIST
      LIST = 'HIGHER APPLIED VOLTAGE THE CHANGE IN CURRENT'
      CALL QCMOV(20,12)
      WRITE(6,805) LIST
      LIST = 'DECREASES TO A POINT WHERE THE CURRENT IS AGAIN'
      CALL QCMOV(20,11)
      WRITE(6,805) LIST
      LIST = 'MEASURED AT INCREMENTS OF THE APPLIED VOLTAGE'
      CALL QCMOV(20,10)
      WRITE(6,805) LIST
      LIST = 'EQUAL TO 2.0 mV'
      CALL QCMOV(20,9)
      WRITE(6,805) LIST
      LIST = 'PRESS ANY KEY TO CONTINUE'
      CALL QCMOV(20,7)
      WRITE(6,805) LIST
      CALL QINKEY(I,M8)
      CALL QCLEAR(0,4)
400   LIST = ' '
      CALL QCMOV(20,20)
      WRITE(6,805) LIST
      LIST = 'THE DATA RECORDED FOR THIS EXPERIMENT EITHER:'
      CALL QCMOV(20,19)
      WRITE(6,805) LIST
      LIST = ' '
      CALL QCMOV(6,18)
      WRITE(6,805) LIST
      LIST = '1. REFLECTS THAT THE CHANGES IN CURRENT'
      CALL QCMOV(20,17)
      WRITE(6,805) LIST
      LIST = 'MEASUREMENT DESCRIBED EARLIER DID NOT OCCUR'
      CALL QCMOV(20,16)
      WRITE(6,805) LIST
      LIST = 'AT A POINT(S), RELATIVE TO ECOOR, THAT WOULD'
      CALL QCMOV(20,15)
      WRITE(6,805) LIST
      LIST = 'GENERATE ENOUGH DATA POINTS FOR THE USE OF THE'
      CALL QCMOV(20,14)
      WRITE(6,805) LIST
      LIST = 'CENTRAL DIFFERENCE METHOD.'
      CALL QCMOV(20,13)
      WRITE(6,805) LIST
```

```
LIST = 'OR'
CALL GCMOV(39,11)
WRITE(6,805) LIST
LIST = '2. THE MODEL 351 DID NOT RECORD THE APPLIED'
CALL GCMOV(20,9)
WRITE(6,805) LIST
LIST = 'POTENTIAL PROPERLY. THIS CAN BE CORRECTED'
CALL GCMOV(20,8)
WRITE(6,805) LIST
LIST = 'IF THE PROBLEM IS NOT CORRECTED THEN THE'
CALL GCMOV(20,7)
WRITE(6,805) LIST
LIST = 'CENTRAL DIFFERNCE METHOD WILL NOT WORK, AND THE'
CALL GCMOV(20,6)
WRITE(6,805) LIST
LIST = 'CUBIC SPLINE METHOD MAY NOT WORK'
CALL GCMOV(20,5)
WRITE(6,805) LIST
LIST = 'PRESS ANY KEY TO CONTINUE'
CALL GCMOV(20,3)
WRITE(6,805) LIST
CALL QINKEY(I,MB)
CALL GCLEAR(0,4)
LIST = 'IF AN ATTEMPT IS MADE TO CORRECT THE'
CALL GCMOV(20,17)
WRITE(6,805) LIST
LIST = 'PROBLEM THE CORRECT METHOD IS DETERMINED'
CALL GCMOV(20,16)
WRITE(6,805) LIST
LIST = 'IF THE ATTEMPT IS UNSUCCESSFUL THEN THE'
CALL GCMOV(20,15)
WRITE(6,805) LIST
LIST = 'CUBIC SPLINE METHOD SHOULD BE USED.'
CALL GCMOV(20,14)
WRITE(6,805) LIST
LIST = 'IF THE ATTEMPT IS SUCCESSFUL EITHER'
CALL GCMOV(20,13)
WRITE(6,805) LIST
LIST = 'METHOD MAY BE USED.'
CALL GCMOV(20,12)
WRITE(6,805) LIST
LIST = 'IF THIS SET OF MESSAGES APPEARS AGAIN'
 CALL GCMOV(20,10)
WRITE(6,805) LIST
LIST = 'AFTER ATTEMPTING TO CORRECT THE PROBLEM'
 CALL GCMOV(20,9)
WRITE(6,805) LIST
LIST = 'THEN THE CURRENT DENSITY VALUES ARE SUCH'
 CALL GCMOV(20,8)
WRITE(6,805) LIST
LIST = 'THAT NEITHER METHOD CAN BE USED'
```

```
            CALL GCMOV(20,7)
            WRITE(6,805) LIST
            LIST = 'PRESS ANY KEY TO CONTINUE'
            CALL GCMOV(20,5)
            WRITE(6,805) LIST
            CALL GINKEY(I,MB)
420         CALL GCLEAR(0,2)
            LIST = '1. DO NOT CORRECT THE PROBLEM.'
            CALL GCMOV(25,13)
            WRITE(6,805) LIST
            LIST = '2. TRY AND CORRECT THE PROBLEM.'
            CALL GCMOV(25,11)
            WRITE(6,805) LIST
            LIST = 'ENTER A 1 OR A 2'
            CALL GCMOV(32,9)
            WRITE(6,805) LIST
            CALL GCMOV(39,8)
            READ(6,810,ERR = 450, IOSTAT = J4) M
450         IF ((M .GT. 2) .OR. (J4 .NE. 0)) THEN
              CALL GCLEAR(0,4)
              LIST = ' '
              CALL GCMOV(21,16)
              WRITE(6,805) LIST
              LIST = 'YOU DID NOT ENTER AN INTEGER OF'
              CALL GCMOV(20,15)
              WRITE(6,805) LIST
              LIST = 'VALUE 1 OR 2'
              CALL GCMOV(37,14)
              WRITE(6,805) LIST
              LIST = 'PRESS ANY KEY TO CONTINUE'
              CALL GCMOV(25,12)
              WRITE(6,805) LIST
              CALL GCMOV(39,11)
              CALL GINKEY(I,MB)
              GOTO 420
            ENDIF
            IF (M .EQ. 1) THEN
              N9 = 1
              GOTO 1000
            ENDIF

C     AN ATTEMPT IS BEING MADE TO CORRECT THE POTENTAIL VALUES SUCH
C     THAT THEY ACCURATELY REFLECT THE EXPERIMENTAL POTENTIALS.
C     BLOCKS 500 AND 600 CORRECT THE PROBLEM, BUT ONLY ONE OF THE
C     TWO BLOCKS ARE USED FOR ANY GIVEN FILE.

            IF (M .EQ. 2) THEN
              DO 500 I = 1,N,2
               Y1 = MDATA(I,1)
               Y2 = MDATA(I+1,1)
               IF (Y1 .EQ. Y2) THEN
```

```
              MDATA(I,1) = MDATA(I,1) - 0.0005D0
              N9 = 1
             ENDIF
500          CONTINUE
             DO 600 I = 2,N,2
              Y1 = MDATA(I,1)
              Y2 = MDATA(I+1,1)
              IF (Y1 .EQ. Y2) THEN
                MDATA(I,1) = MDATA(I,1) - 0.0005D0
               N9 = 1
              ENDIF
600          CONTINUE
             IF (N9 .EQ. 1) THEN
                CALL GCLEAR(0,2)
                LIST = ' '
                CALL GCMOV(30,14)
                WRITE(6,805) LIST
                LIST = 'CORRECTION SUCCESSFUL'
                CALL GCMOV(30,13)
                WRITE(6,805) LIST
                LIST = 'USE EITHER METHOD'
                CALL GCMOV(30,11)
                WRITE(6,805) LIST
                LIST = 'PRESS ANY KEY TO CONTINUE'
                CALL GCMOV(30,9)
                WRITE(6,805) LIST
                CALL GINKEY(I,M8)
                GOTO 10
             ELSE
                CALL GCLEAR(0,4)
                LIST = 'CORRECTION UNSUCCESSFUL'
                CALL GCMOV(30,13)
                WRITE(6,805) LIST
                LIST = 'USE EITHER CUBIC SPLINE METHOD'
                CALL GCMOV(30,11)
                WRITE(6,805) LIST
                LIST = 'PRESS ANY KEY TO CONTINUE'
                CALL GCMOV(30,9)
                WRITE(6,805) LIST
                CALL GINKEY(I,M8)
                GOTO 1000
             ENDIF
            ENDIF
      ENDIF
805   FORMAT(A50,\)
810   FORMAT (I1)
1000  END
```

```
*     XDAT  : THE X VALUES OF THE FIRST GRAPH
*     YDAT  : THE Y VALUES OF THE FIRST GRAPH
*     XDAT1 : THE X VALUES OF THE SECOND GRAPH
*     YDAT1 : THE Y VALUES OF THE SECOND GRAPH
*     TX    : XDAT AS ORIGINAL PASSED AND SCALED. USED TO ALLOW
*             THE USER TO RESCALE THE AXIS AND MAINTAIN ALL THE
*             VALUES ORIGINALLY PASSED
*     TY    : USED AS TX BUT FOR YDAT
*     TX1   : USED AS TX BUT FOR XDAT1
*     TY1   : USED AS TX BUT FOR YDAT1

C     IN THIS SUBROUTINE ALL PROMPTS ARE REPLIED TO BY THE USE OF
C     SOFT KEYS. THIS MEANS THAT A CARRIAGE RETURN IS NOT USED.

      SUBROUTINE GRAPH1(XDAT,YDAT,N3,J,NAME,XDAT1,YDAT1,N4,YNAME,TITLE2,
     +A1,A2,C1,C2,TITLE3)
      REAL*4 XDAT(2000),YDAT(2000),XDAT1(2000),YDAT1(2000),F,F1,F2
      REAL*4 TX(2000),TY(2000),TX1(2000),TY1(2000),X2(1),Y2(1)
      INTEGER*2 K2,K3,K5,K6,L,XL,XR,YL,YR,MB,A1,A2,C1,C2,M,IA(80)
      INTEGER*2 I,J,N,K7,K8,K9, JCOL(10),JROW(10),N1,NB,N3,N4,N5,N6,N7
      INTEGER*2 L1,L2,L3,L4
      CHARACTER NAME*8,YNAME*5,FACT*6,XNAME*12,TITLE*15,TITLE2*32
      CHARACTER*28, QUES, TITLE3*50
      N7 = 1
      M = 0
      N = MAX(N3,N4)
      N3 = N3 - 3
      N4 = N4 - 3
      IF (N4 .LE. 0) N4 = 1

C     THE ARRAYS ARE SCALED SUCH THAT THE Y AXIS VALUES ARE DISCERNIBLE
C     FROM EACH OTHER

      F = MIN(ABS(YDAT(1)),ABS(YDAT(N3)))
      F1 = MIN(ABS(YDAT1(1)),ABS(YDAT1(N4)))
      IF (N4 .GT. 2) THEN
         F2 = MIN(F1,F)
      ELSE
         F2 = F
      ENDIF
      IF (F2 .GE. 1.) THEN
        F2 = 1.0
        FACT = 'x 1'
      ENDIF
      IF ((F2 .GE. .1) .AND. (F2 .LT. 1.)) THEN
        F2 = 10
        FACT = 'x 10'
      ENDIF
```

```
      IF ((F2 .GE. .01) .AND. (F2 .LT. .1)) THEN
        F2 = 100.
        FACT = 'x 100'
      ENDIF
      IF ((F2 .GE. .001) .AND. (F2 .LT. .01)) THEN
        F2 = 1000
        FACT = 'x 1000'
      ELSE
         IF (F2 .LE. 0.0010) THEN
         F2 = 1000
         FACT = 'x 1000'
         ENDIF
      ENDIF
      DO 6 I = 1,2000
         YDAT(I) = F2*YDAT(I)
         YDAT1(I) = F2*YDAT1(I)
6     CONTINUE
5     N5 = N3
      N6 = N4

C     THE PERMANENT ARRAYS ARE FILLED

      DO 10 I = 1,2000
         TX(I) = XDAT(I)
         TY(I) = YDAT(I)
         TX1(I) = XDAT1(I)
         TY1(I) = YDAT1(I)
10    CONTINUE

C     THE GRAPH AXIS AND WINDOW ARE DETERMINED

      XMAX = -1000
      XMIN = 1000
      YMAX = XMAX
      YMIN = XMIN
      XMAX1 = XMAX
      XMIN1 = XMIN
      YMAX1 = XMAX
      YMIN1 = XMIN
      JCOL1 = 90
      JCOL2 = 610
      JROW1 = 35
      JROW2 = 300
      DO 200 I = 1,N3
      YMAX = MAX(YDAT(I),YMAX)
      XMAX = MAX(XDAT(I),XMAX)
      XMIN = MIN(XDAT(I),XMIN)
      YMIN = MIN(YDAT(I),YMIN)
200   CONTINUE
205   DO 210 I = 1,N4
      XMAX1 = MAX(XDAT1(I),XMAX1)
```

```
      YMAX1 = MAX(YDAT1(I),YMAX1)
      XMIN1 = MIN(XDAT1(I),XMIN1)
      YMIN1 = MIN(YDAT1(I),YMIN1)
210   CONTINUE
215   YFIN = (MAX(YMAX,YMAX1)) + .1
      YST  = (MIN(YMIN,YMIN1)) - .1
      XST  = (MIN(XMIN,XMIN1)) -.1
      XFIN = (MAX(XMAX,XMAX1)) + .1
225   YMIN = YST
      YMAX = YFIN
      XMAX = XFIN
      XMIN = XST
      YORG = YMIN
      XORG = XMIN
      XNAME = 'LOG I A/cm2'
      TITLE = 'POTENTIODYNAMIC'
      IOPT = 0
      YOVERX = 0.3
      ASPECT = 1.5

C     THE GRAPH AXIS ARE DRAWN AND LABELED. TITLES ARE PLACED

      CALL GSMODE(16)
      CALL GPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,XORG,YORG,
     +IOPT,YOVERX,ASPECT)
       XMAJOR = (XFIN-XST)/10.
       XMINOR = 0
       LABEL = -1
       NDEC  = 2
       YMAJOR =(YFIN-YST)/10.
       YMINOR = 0
       NKNT = 5
       NDOTS = 0
       IKOLOR = 1
       ISYMBOL=-2
       KLRSYM=1
       CALL GCOLOR(0,0)
       CALL GSETUP(NDOTS,IKOLOR,ISYMBOL,KLRSYM)
       CALL GXAXIS(XST,XFIN,XMAJOR,XMINOR,LABEL,NDEC)
       CALL GYAXIS(YST,YFIN,YMAJOR,YMINOR,LABEL,NDEC)
       CALL GXTICS(NKNT,JCOL,JROW)
       CALL GGTXT(12,XNAME,2,340,1,0)
       CALL GYTICS(NKNT,JCOL,JROW)
       CALL GGTXT(5,YNAME,2,1,246,-1)
       CALL GGTXT(6,FACT,2,1,162,-1)
       CALL GGTXT(15,TITLE,2,254,335,0)
       CALL GGTXT(8,NAME,2,382,335,0)
       CALL GGTXT(32,TITLE2,2,254,321,0)
       CALL GGTXT(50,TITLE3,2,182,307,0)
       IKOLOR = 2
       CALL GSETUP(NDOTS,IKOLOR,ISYMBOL,KLRSYM)
```

```
      CALL GRTOI(XST,YST,L1,L2)
      CALL GRTOI(XFIN,YFIN,L3,L4)
      CALL GLINE(L1,L2,L1,L4,2)
      CALL GLINE(L1,L4,L3,L4,2)
      CALL GLINE(L3,L4,L3,L2,2)
      CALL GLINE(L3,L2,L1,L2,2)

C     THE FIRST CURVE IS DRAWN. THE LINE COLOR IS CHANGED AND THE
C     SECOND CURVE IS DRAWN

      CALL GTABL(1,N5,XDAT,YDAT)
      IKOLOR = 1
      CALL GSETUP(NDOTS,IKOLOR,ISYMBOL,KLRSYM)
      CALL GTABL(1,N6,XDAT1,YDAT1)
      CALL GBEEP
      M = M + 1

C     IF THE CURVES ARE OF DERIVATIVES AND A LINEAR REGION HAS ALREADY
C     BEEN SELECTED, THE USER MAY DECIDE TO GENERATE THE TAFEL CONSTANTS
C     BASED ON THIS PORTION OF THE CURVE.

380      IF (( M .GT. 1) .AND. (N4 .GE.0)) THEN
            QUES = 'GENERATE TAFEL CONSTANT'
            CALL GGTXT(28,QUES,3,1,335,0)
            QUES = '      Y OR N ?'
            CALL GGTXT(28,QUES,3,1,320,0)
            CALL GBEEP
            CALL GINKEY(I,M8)
            IF ((M8 .EQ. 89) .OR. (M8 .EQ. 121)) THEN
               QUES = ' '
               CALL GGTXT(28,QUES,3,1,335,0)
               CALL GGTXT(28,QUES,3,1,320,0)
381            QUES = 'IS THIS THE ANODIC OR'
               CALL GGTXT(28,QUES,3,1,335,0)
               QUES = 'CATHODIC BRANCH'
               CALL GGTXT(28,QUES,3,1,320,0)
               QUES = '      A OR C ?'
               CALL GGTXT(28,QUES,3,1,305,0)
               CALL GBEEP
               CALL GINKEY(I,M8)
               IF ((M8 .EQ. 65) .OR. (M8 .EQ.97)) THEN
                  A1 = K8
                  A2 = K9
                  GOTO 390
               ENDIF
               IF ((M8 .EQ. 67) .OR. (M8 .EQ. 99)) THEN
                  C1 = K6
                  C2 = K7
                  GOTO 390
               ELSE
                  GOTO 381
```

```
            ENDIF
          ENDIF
          IF ((M8 .EQ. 78) .OR. (M8 .EQ. 110)) THEN
            GOTO 390
          ELSE
            GOTO 380
          ENDIF
        ENDIF

C     BY THE USE OF THE CURSOR KEYS ON THE NUMERIC KEYPAD THE USER
C     SELECTS A NEW SET OF AXIS TO BE PLOTTED

390    QUES = ' '
       CALL QGTXT(28,QUES,3,1,335,0)
       CALL QGTXT(28,QUES,3,1,320,0)
       CALL QGTXT(28,QUES,3,1,305,0)
       QUES = 'NEW AXIS Y OR N ?'
       CALL QGTXT(28,QUES,3,1,335,0)
       CALL QBEEP
       CALL QINKEY(I,M8)
       IF ((M8 .EQ. 89) .OR. (M8 .EQ. 121)) THEN
        GOTO 400
       ELSE
         IF ((M8 .EQ. 78) .OR. (M8 .EQ. 110)) THEN
           GOTO 800
         ELSE
           GOTO 390
         ENDIF
       ENDIF
400    QUES = ' '
       CALL QGTXT(28,QUES,1,1,335,0)
       CALL QGTXT(28,QUES,1,1,320,0)
       QUES = 'NEW,LAST,OR ORIGINAL (N,L,O)'
       CALL QGTXT(28,QUES,3,1,335,0)
       CALL QBEEP
       CALL QINKEY(I,M8)
        IF ((M8 .EQ. 79) .OR. (M8 .EQ. 111)) THEN
          DO 410 I = 1,N3
            XDAT(I) = TX(I)
            YDAT(I) = TY(I)
410       CONTINUE
          DO 420 I = 1,N4
            XDAT1(I) = TX1(I)
            YDAT1(I) = TY1(I)
420       CONTINUE
          M = 0
          GOTO 5
        ENDIF
        IF ((M8 .EQ. 76) .OR. (M8 .EQ. 108)) THEN
         XST = XST1
         XFIN = XFIN1
```

```
        YST = YST1
        YFIN = YFIN1
        N5 = M5
        N6 = M6
        GOTO 530
       ENDIF
       IF (( M8 .EQ. 78) .OR. (M8 .EQ. 110)) THEN
         XST1 = XST
         YST1 = YST
         XFIN1 = XFIN
         YFIN1 = YFIN
         M5 = N5
         M6 = N6
430      QUES = 'USE THE CURSOR KEY TO MOVE'
         CALL QGTXT(28,QUES,3,1,335,0)
         QUES = 'TO THE LOWER LEFT HAND'
         CALL QGTXT(28,QUES,3,1,320,0)
         QUES = 'CORNER, HOME TO EXIT'
         CALL QGTXT(28,QUES,3,1,305,0)
         CALL QBEEP
         CALL QLSTAT(1)
         CALL QLTYPE(3,3)
         CALL QLMOV(320,175)
500      CALL QINKEY(I,L)
          IF (L .EQ. 72) THEN
           CALL QLMOVR(0,5)
           GOTO 500
          ENDIF
          IF (L .EQ. 75) THEN
           CALL QLMOVR(-5,0)
           GOTO 500
          ENDIF
          IF (L .EQ. 77) THEN
           CALL QLMOVR(5,0)
           GOTO 500
          ENDIF
          IF (L .EQ. 80) THEN
           CALL QLMOVR(0,-5)
           GOTO 500
          ENDIF
          IF (L .EQ. 71) THEN
           CALL QLREAD(XL,YL)
           CALL QLMOVR(320-XL,175-YL)
           CALL QITOR(XL,YL,XST,YST)
           X2(1) = XST
           Y2(1) = YST
           CALL QSETUP(5,3,43,3)
           CALL QTABL(0,N7,X2,Y2)
           GOTO 510
          ELSE
           GOTO 430
```

```
          ENDIF
510       QUES = 'USE THE CURSOR KEY TO MOVE'
          CALL QGTXT(28,QUES,3,1,335,0)
          QUES = 'TO THE UPPER RIGHT HAND'
          CALL QGTXT(28,QUES,3,1,320,0)
          QUES = 'CORNER, HOME TO EXIT'
          CALL QGTXT(28,QUES,3,1,305,0)
          CALL QBEEP
520       CALL QINKEY(I,L)
          IF (L .EQ. 72) THEN
           CALL QLMOVR(0,5)
           GOTO 520
          ENDIF
          IF (L .EQ. 75) THEN
           CALL QLMOVR(-5,0)
           GOTO 520
          ENDIF
          IF (L .EQ. 77) THEN
           CALL QLMOVR(5,0)
           GOTO 520
          ENDIF
          IF (L .EQ. 80) THEN
           CALL QLMOVR(0,-5)
           GOTO 520
          ENDIF
          IF (L .EQ. 71) THEN
           CALL QLREAD(XR,YR)
           CALL QLMOVR(320-XR,175-YR)
           CALL QITOR(XR,YR,XFIN,YFIN)
           X2(1) = XFIN
           Y2(1) = YFIN
           CALL QTABL(0,N7,X2,Y2)
           CALL QLSTAT(0)
           GOTO 530
          ELSE
           GOTO 510
           ENDIF
          ENDIF
530       IF (YNAME .EQ. 'VOLTS') THEN
             GOTO 780
          ENDIF
             K6 = 1
             K7 = N3
          DO 600 I = 1,N3
            X = TX(I)
            Y=TY(I)
            IF (((X .GT. XST) .AND. (X .LT. XFIN)) .AND. ((Y .GT. YST)
     +      .AND. (Y .LT. YFIN))) THEN
              K6 = I
              GOTO 605
            ENDIF
```

```
600         CONTINUE
605         DO 610 I = K6,N3
              X = TX(I)
              Y = TY(I)
              IF (((X .LT. XST) .OR. (X .GT. XFIN)) .OR. ((Y .LT. YST)
     +        .OR. (Y .GT. YFIN))) THEN
                 K7 = I
                 GOTO 630
              ENDIF
610         CONTINUE
630         N5 = K7-K6
            DO 670 I = K6,K7
             YDAT(I-K6+1) = TY(I)
             XDAT(I-K6+1) = TX(I)

670         CONTINUE
            K8 = 1
            K9 = N4
            K2 = N4
            K3 = 1
            DO 700 I = 1,N4
              X = TX1(I)
              Y = TY1(I)
              IF (((X .GT. XST) .AND. (X .LT. XFIN)) .AND. ((Y .GT. YST)
     +        .AND. (Y .LT. YFIN))) THEN
                K8 = I
                GOTO 715
              ENDIF
700         CONTINUE
715         DO 720 I = K8,N4
              X = TX1(I)
              Y = TY1(I)
              IF (((X .LT. XST) .OR. (X .GT. XFIN)) .OR. ((Y .LT. YST)
     +        .OR. (Y .GT. YFIN))) THEN
                 K9 = I
                 GOTO 730
              ENDIF

720         CONTINUE
730         DO 770 I = K8,K9
             YDAT1(I-K8+1) = TY1(I)
             XDAT1(I-K8+1) = TX1(I)
770         CONTINUE
            N6 = K9 - K8
            GOTO 225
780         N5 = 0
            N6 = 0
            DO 785 I = 1,N3
              X = TX(I)
              Y=TY(I)
              IF (((X .GT. XST) .AND. (X .LT. XFIN)) .AND. ((Y .GT. YST)
```

```
     +       .AND. (Y .LT. YFIN))) THEN
                N5 = N5 +1
                XDAT(N5) = TX(I)
                YDAT(N5) = TY(I)
              ENDIF
785         CONTINUE
            DO 790 I = 1,N4
              X = TX1(I)
              Y=TY1(I)
              IF (((X .GT. XST) .AND. (X .LT. XFIN)) .AND. ((Y .GT. YST)
     +       .AND. (Y .LT. YFIN))) THEN
                N6 = N6 +1
                XDAT1(N6) = TX1(I)
                YDAT1(N6) = TY1(I)
              ENDIF
790         CONTINUE
            M = 0
            GOTO 225
800       QUES = 'DO YOU WANT TO PLOT'
          CALL QGTXT(28,QUES,3,1,335,0)
          QUES = 'THIS GRAPH Y OR N ?'
          CALL QGTXT(28,QUES,3,1,320,0)
          CALL QBEEP
          CALL QINKEY(I,MB)
          IF ((MB .EQ. 89) .OR. (MB .EQ. 109)) THEN
              GOTO 830
          ELSE
             IF ((MB .EQ. 78) .OR. (MB .EQ. 110)) THEN
               GOTO 850
             ELSE
               GOTO 800
             ENDIF
          ENDIF
830       CALL PLOT(XDAT,YDAT,N5,XDAT1,YDAT1,N6,XNAME,YNAME,TITLE,
         +XMAX,XMIN,YMAX,YMIN,XST,XFIN,YST,YFIN,XMAJOR,YMAJOR,TITLE2,
         +FACT,NAME)
          IF (YNAME .EQ. 'VOLTS') THEN
             GOTO 390
          ENDIF
          IF (YNAME .EQ.  'Dv/Di') THEN
             GOTO 380
          ENDIF
850       DO 860 I = 1,2000
           TX(I)=0
           TY(I)=0
           TX1(I) = 0
           TY1(I) = 0
           XDAT(I) = 0
           XDAT1(I) = 0
           YDAT(I) = 0
           YDAT1(I) = 0
```

```
860     CONTINUE
        CALL GSMODE(16)
        CALL GSMODE(3)
900     FORMAT(A1)
        END
```

```
*     THIS SUBROUTINE PLOTS THE GRAPHS AS SHOWN ON THE SCREEN DURING
*     THE CALLING SUBROUTINE GRAPH1.
*     THE MAJORITY OF THE PROGRAM IS INVOLVED IN SETTING UP ARGUEMENTS
*     TO BE PASSED TO SUBROUTINE CONTAINED WITHIN THE PLOTMATICS
*     LIBRARY. THIS MANUAL SHOULD BE CONSULTED IN ORDER TO DETERMINE
*     THE USE OF THE ARGUEMENTS OTHER THAN THOSE DEFINED BELOW
*
*     XDAT  : THE ARRAY CONTAINING THE X VALUES OF THE FIRST CURVE
*             TO BE PLOTTED.
*     YDAT  : THE FIRST CURVE'S Y VALUES
*     XDAT1 : THE ARRAY CONATINING THE Y VALUES OF THE SECOND CURVE
*             TO BE PLOTTED.
*     YDAT1 : THE SECOND CURVE'S Y VALUES
*     TITLE : THE TITLE OF THE PLOT
*     TITLE2: AN DISCRIBTION OF THE GRAPH
*     XNAME : THE X AXIS TITLE
*     SAMPLE: THE Y AXIS TITLE PROPERLY POSITION AND CATENTATED

      SUBROUTINE PLOT(XDAT,YDAT,NPT1,XDAT1,YDAT1,NPT2,XNAME,YNAME
     +,TITLE,XMAX,XMIN,YMAX,YMIN,XST,XFIN,YST,YFIN,XMAJOR,YMAJOR
     +,TITLE2,FACT, NAME)
      REAL*4 XDAT(2000),YDAT(2000),XDAT1(2000),YDAT1(2000)
      INTEGER*2 JCOL(10),JROW(10),NPT1,NPT2
      INTEGER I3,I4,I5,I6,I7,I8,I11,I12,MB
      CHARACTER*8 NAME,YNAME*5,XNAME*11,TITLE*15,SAMPLE*12,TITLE2*32
     +, FACT*6,A4*1, QUES*28
      A4 = ' '
        SAMPLE = YNAME//A4//FACT
        XORG = XST
        YORG = YST
        DTX = ABS(XMAX-XMIN)
        DX = DTX /MAX(NPT1,NPT2)
        DTY = ABS(YMAX-YMIN)
        IOPT = 0
        NKNT = 5
        NDOTS = 0
        LABEL = -1
        NDEC = 3
        ISIZE = 1
        ISTEP = 2
        IPORT = 1
        CALL ZINIT(ISIZE,ISTEP,IPORT,BATCH)
        JCL = 20
        JCR = 620
        ICL = 10
        ICR = 320
        I3 = 12
        I4 = 150
```

```
I5 = 236
I6 = 336
I7 = 340
I8 = 322
I11 = 386
I12 = 182
CALL ZCONV(JCL,ICL,16)
CALL ZCONV(JCR,ICR,16)
CALL ZCONV(I3,I4,16)
CALL ZCONV(I5,I6,16)
CALL ZCONV(I7,I8,16)
CALL ZCONV(I11,I12,16)
XMIN1 =XMIN -.2*DTX
XMAX1 = XMAX+  .1*(DTX)
YMAX1 = YMAX + .1*DTY
YMIN1 = YMIN - .1*DTY
NDEC = 2
CALL ZPLOT(JCL,JCR,ICL,ICR,XMIN1,XMAX1,YMIN1,YMAX1,XORG,YORG)
CALL ZRTOI(XST,YST,L1,L2)
CALL ZRTOI(XFIN,YFIN,L3,L4)
CALL ZSETUP(-1,1,0,1)
MINOR = 1
CALL ZXAXIS(XST,XFIN,XMAJOR,MINOR,LABEL,NDEC)
CALL ZYAXIS(YST,YFIN,YMAJOR,MINOR,LABEL,NDEC)
CALL ZXTICS(10,JCOL,JROW)
CALL ZYTICS(10,JCOL,JROW)
CALL ZLINE(L1,L2,L1,L4,1)
CALL ZLINE(L1,L4,L3,L4,1)
CALL ZLINE(L3,L4,L3,L2,1)
CALL ZLINE(L3,L2,L1,L2,1)
CALL ZPU
QUES = ' '
CALL QGTXT(28,QUES,3,1,335,0)
CALL QGTXT(28,QUES,3,1,320,0)
CALL QGTXT(28,QUES,3,1,305,0)
QUES = 'IF DESIRED INSERT A NEW PEN'
CALL QGTXT(28,QUES,3,1,335,0)
QUES = 'THEN PRESS ANY KEY'
CALL QGTXT(28,QUES,3,1,320,0)
CALL QBEEP
CALL QINKEY(I,M8)
CALL ZTABL(1,NPT1,XDAT,YDAT)
CALL ZPU
QUES = 'IF DESIRED INSERT A NEW PEN'
CALL QGTXT(28,QUES,3,1,335,0)
QUES = 'THEN PRESS ANY KEY'
CALL QGTXT(28,QUES,3,1,320,0)
CALL QBEEP
CALL QINKEY(I,M8)
CALL ZTABL(1,NPT2,XDAT1,YDAT1)
CALL ZPU
```

```
CALL QINKEY(I,MB)
QUES = 'IF DESIRED INSERT A NEW PEN'
CALL QGTXT(28,QUES,3,1,335,0)
QUES = 'THEN PRESS ANY KEY'
CALL QGTXT(28,QUES,3,1,320,0)
CALL QBEEP
CALL ZPTXTA(11,XNAME,1)
CALL ZSETUP(2,1,0,1)
CALL ZIW(1,1,1760,1400,0)
CALL ZPTXT(15,TITLE,1,I5,I6)
CALL ZPTXT(8,NAME,1,I11,I6)
CALL ZPTXT(32,TITLE2,1,I5,I8)
CALL ZEXTXT(0,0,0,0,0,1)
CALL ZPU
CALL ZPA(I3,I4)
CALL ZLABEL(12,SAMPLE)
CALL ZFINIS
END
```

```
*     THIS IS THE FIRST SUBROUTINE CALLED WHEN USING THE CUBIC
*     SPLINE METHOD.
*
*     MDATA : THE ARRAY CONTAINING POTENTIAL AND LOG CURRENT DENSITY
*     XDAT  : THE ARRAY CONTAINING THE LOG CURRENT VALUES FOR THE
*             CATHODIC BRANCH
*     YDAT  : THE ARRAY CONTAINING THE DERIVATIVES FOR THE CATHODIC
*             BRANCH
*     XDAT1 : SAME A XDAT BUT FOR THE ANODIC BRANCH
*     YDAT1 : SAME AS YDAT BUT  FOR THE ANODIC BRANCH
*     DIV   : THE ARRAY CONTAINING THE DERIVATIVES WHICH IS BEING
*             RETURN FROM SUBSEQUENT CALLED SUBROUTINES
*
*     THE INTEGER VARIABLES ACTUALLY USED IN THIS SUBROUTINE ARE
*     USED TO DEFINE THE REGIONS WITHIN BOTH BRANCHES AVAILABLE
*     FOR USE
*

      SUBROUTINE SLOPE1(MDATA,N,J,NAME,A1,A2,C1,C2)
      REAL*4 XDAT(2000),YDAT(2000),XDAT1(2000),YDAT1(2000)
      REAL*8 MDATA(2000,2),DIV(1000)
      INTEGER*2 I,J,N,K,MI,N3,N4,N1,K1,A1,A2,C1,C2
      CHARACTER NAME*8,YNAME*5,TITLE2*32,TITLE3*50
      N1 = N
      YNAME = 'Dv/Di'
      TITLE2 = 'TAFEL SLOPES: CUBIC SPLINE'
      TITLE3=' '

C     THE CATHODIC BRANCH IS CHECKED FOR AN INCREASE IN LOG
C     CURRENT DENSITY WHICH COULD CAUSE THE CALCULATION OF A
C     DERIVATIVE OF INFINTE SLOPE

      DO 20 I = J-1,2,-1
         IF (MDATA(I-1,2) .LE. MDATA(I,2)) GOTO 25
20    CONTINUE
      K = MAX((J-250),1)
      GOTO 26
25    K = I
26    MI = J-9
      J1 = K
      N3 = ABS(K-MI)
      M = 7
      CALL CSPLIN(MDATA,N,M,DIV,K,N3)
      DO 100 I = 1,N3-3
      XDAT(I) = MDATA(J-4-I,2)
      YDAT(I) = DIV(N3-I+1)
100   CONTINUE
```

```
C     THE ANODIC BRANCH IS CHECKED FOR LOG CURRENT VALUES WHICH
C     COULD CAUSE THE CALCULATION OF A DERIVATIVE OF INFINITE SLOPE

      DO 120 I = J+1,N1
         IF (MDATA(I,2) .LE. MDATA(I-1,2)) GOTO 125
120   CONTINUE
      K = MIN((J+250),(N1))
      GOTO 126
125   K = I
126   J1 = J
      MI = K - 9
      N4 = ABS(MI-J)
      CALL CSPLIN(MDATA,N,M,DIV,J,N4)
      DO 200 I = 1,N4
      XDAT1(I) = MDATA(J1+I+4,2)
      YDAT1(I) = DIV(I)
200   CONTINUE
      CALL GRAPH1(XDAT,YDAT,N3,J,NAME,XDAT1,YDAT1,N4,YNAME,TITLE2,
     +A1,A2,C1,C2,TITLE3)
      END
```

```
*     THIS SUBROUTINE SETS UP THE MATRIX OF COEFFICIENTS TO
*     BE SOLVED IN ORDER TO DETERMINE THE DERIVATIVE USING THE
*     CUBIC SPLINE METHOD.
*
*     MDATA : THE ARRAY CONTAINING POTENTIAL AND LOG CURRENT DENSITY
*     CMAT  : THE 7x7 MATRIX CONTAINING THE COEFFICIENTS
*     YMAT  : THE 7x1 VECTOR CONTAINING THE RIGHT HAND SIDE OF
*             THE SYSTEM OF EQUATIONS. WHEN RETURNED FROM THE
*             SOLVING SUBROUTINE IT CONTAINS THE SOLUTION.

      SUBROUTINE CSPLINE(MDATA,J,M,DIV,K,MI)
      REAL*8 XDAT(1000), YDAT(1000),MDATA(2000,2),YMAT(7)
      REAL*8 CMAT(7,7),AI,BI,CI
      REAL*8 DIV(1000)
      INTEGER*2 N,MI,K,J
      DO 10 I = 1,7
        DO 5 J = 1,7
          CMAT(I,J) = 0.DO
5       CONTINUE
10    CONTINUE
      M = 7
      K = K-1
      DO 200 N = 1,MI
20    K = K + 1
      CMAT(1,1) = 2.DO*(MDATA(K+2,2)-MDATA(K,2))
      CMAT(1,2) = MDATA(K+2,2) - MDATA(K+1,2)
      CMAT(7,7) = 2.DO*(MDATA(K+8,2)-MDATA(K+6,2))
      CMAT(7,6) = MDATA(K+8,2)-MDATA(K+7,2)
      YMAT(1)=    ((MDATA(K+2,1)-MDATA(K+1,1))/(MDATA(K+2,2)-MDATA
     +(K+1,2))) - ((MDATA(K+1,1)-MDATA(K,1))/(MDATA(K+1,2)-MDATA
     +(K,2)))
      YMAT(1) = YMAT(1)*6.
      YMAT(7)=   ((MDATA(K+8,1)-MDATA(K+7,1))/(MDATA(K+8,2)-MDATA
     +(K+7,2))) - ((MDATA(K+7,1)-MDATA(K+6,1))/(MDATA(K+7,2)-MDATA
     +(K+6,2)))
      YMAT(7) = YMAT(7)*6.
      DO 100 I = 2,6
        K = K +1
        CMAT(I,I) = 2.DO*((MDATA(K+2,2)-MDATA(K,2)))
        CMAT(I,I+1) = MDATA(K+2,2)-MDATA(K+1,2)
        CMAT(I,I-1) = MDATA(K+1,2) - MDATA(K,2)
        YMAT(I) = ((MDATA(K+2,1)-MDATA(K+1,1))/CMAT(I,I+1))
     +-((MDATA(K+1,1)-MDATA(K,1))/CMAT(I,I-1))
        YMAT(I) = YMAT(I)*6.
100   CONTINUE
      K = K - 5
      CALL LINSY1(CMAT,YMAT,M)
      AI = YMAT(5) - YMAT(4)
```

```
      AI = AI/((6.DO)*((MDATA(K+4,2)-MDATA(K+3,2))))
      BI = YMAT(4)/2.DO
      CI = (MDATA(K+4,1)-MDATA(K+3,1))/(MDATA(K+4,2)-MDATA(K+3,2))
      CI = CI-(1/6)*(((2*YMAT(4))+YMAT(5))*(MDATA(K+4,2)-MDATA(K+
     +3,2)))
      DIV(N) = CI
200   CONTINUE
      END
```

```
*     THIS SUBROUTINE SOLVES THE 7x7 SYSTEM OF EQUATIONS GENERATED
*     BY CSPLIN.FOR.  THE PROGRAM WAS ORIGINALLY FROM THE TEXTBOOK
*     "FORTRAN 77 for Engineers and Scientists" BY NYHOFF AND
*     LEESTMA, PAGES 264-265. THE MAIN MODIFICATION IS THE ELIMINATION
*     OF THE VARIABLE DIMENSIONING OF ARRAYS PRESENT IN THE ORIGINAL
*     PROGRAM.

      SUBROUTINE LINSY1(COMAT,YMAT,N)
      INTEGER N,PIVOT
      REAL*8 AUG(7,8), COMAT(7,7), YMAT(7),X(7),MULT,TEMP
      DO 20 I = 1, N
        DO 10 J = 1,N
           AUG(I,J) = COMAT(I,J)
10      CONTINUE
20    CONTINUE
      DO 30 I = 1,N
           AUG(I,N+1) = YMAT(I)
30    CONTINUE
      DO 70 I = 1,N-1
        IF (AUG(I,I) .EQ. 0.) THEN
          PIVOT = 0
          J = I+1
35          IF ((PIVOT .EQ. 0) .AND. (J .LE. N)) THEN
             IF (AUG(J,I) .NE. 0.) PIVOT = J
             J = J+1
             GOTO 35
            ENDIF
            IF (PIVOT .EQ. 0) THEN
              PRINT*, 'SINGULAR'
              STOP
            ELSE
             DO 40 J = 1,N+1
              TEMP = AUG(I,J)
              AUG(I,J) = AUG(PIVOT,J)
              AUG(PIVOT,J) = TEMP
40           CONTINUE
            ENDIF
          ENDIF
          DO 60 J = I+1,N
            MULT = -AUG(J,I)/AUG(I,I)
            DO 50 K = I,N+1
              AUG(J,K) = AUG(J,K) + MULT*AUG(I,K)
50          CONTINUE
60        CONTINUE
70      CONTINUE
        X(N) = AUG(N,N+1)/AUG(N,N)
        DO 90 J = N-1,1,-1
          X(J) = AUG(J,N+1)
```

```
            DO 80 K = J+1,N
              X(J) = X(J)-(AUG(J,K)*X(K))
80          CONTINUE
            X(J) = X(J)/AUG(J,J)
90        CONTINUE
          DO 100 I = 1,N
           YMAT(I) = X(I)
100       CONTINUE
          END
```

```
*     THIS SUBROUTINE CONVERTS A NUMERIC VARIBLE (X6) OF FORMAT
*     +/- XX.XXXX TO A CHARACTER VARIBLE WHOSE WIDTH IS 8 CHARACTERS.
*     THIS IS NECESSARY AS THE PLOTMATICS LIBRARY IS NOT EQUIPPED WITH
*     A SUBROUTINE FOR PLOTTING NUMERIC VARIBLES.


      SUBROUTINE CONV(A7,X6)
      INTEGER O1,O2,O3,O4,O5,O6
      CHARACTER*1 A6(0:9), C1,C2,C3,C4,C5,C6,C7,C8,A7*8
      A6(0) = '0'
      A6(1) = '1'
      A6(2) = '2'
      A6(3) = '3'
      A6(4) = '4'
      A6(5) = '5'
      A6(6) = '6'
      A6(7) = '7'
      A6(8) = '8'
      A6(9) = '9'
      C1 = ' '
      C2 = ' '
      C3 = ' '
      C4 = ' '
      C5 = ' '
      C6 = ' '
      C7 = ' '
      IF (X6 .LT. 0.0) C1 = '-'
      X6 = ABS(X6)
      X1 = X6/10.
      O1 = IFIX(X1)
      X2 = X6-(10.*O1)
      O2 = IFIX(X2)
      X3 = X6 - (10.*O1) - O2
      X3 = X3/.1
      O3 = IFIX(X3)
      X4 = (X3-O3)/.1  + .001
      O4 = IFIX(X4)
      X5 = (X4-O4)/.1
      O5 = IFIX(X5)
      X7 = (X5-O5)/.1
      O6 = IFIX(X7)
      DO 10 I = 0,9
         IF (O1 .EQ. I) C2 = A6(I)
         IF (O2 .EQ. I) C3 = A6(I)
         IF (O3 .EQ. I) C4 = A6(I)
         IF (O4 .EQ. I) C5 = A6(I)
         IF (O5 .EQ. I) C6 = A6(I)
```

```
      IF (O6 .EQ. I) C7 = A6(I)
10    CONTINUE
      C8 = '.'
      A7 = C1//C2//C3//C8//C4//C5//C6//C7
      RETURN
      END
```

```
*     THIS SUBROUTINE SETS SOME OF THE CHARACTER STRINGS USED
*     WHEN USING THE CENTRAL DIFFERENCE METHOD

      SUBROUTINE SLOPE(MDATA,N,J,NAME,A1,A2,C1,C2,N1,N2,N3,N4)
      REAL*4 XDAT(2000),YDAT(2000),XDAT1(2000),YDAT1(2000)
      REAL*8 MDATA(2000,2)
      INTEGER*2 I,J,N,A1,A2,C1,C2
      CHARACTER NAME*8,YNAME*5,TITLE2*32, LIST*50,TITLE3*50
      YNAME = 'Dv/Di'
      TITLE2 = 'TAFEL SLOPES: CENTRAL DIFFERENCE'
      TITLE3=' '
      CALL DATADEL(MDATA,J,YDAT,YDAT1,XDAT,XDAT1,N2,N1)
      CALL GRAPH1(XDAT,YDAT,N3,J,NAME,XDAT1,YDAT1,N4,YNAME,TITLE2,
     +A1,A2,C1,C2,TITLE3)
      END
```

```
*     THIS SUBROUTINE CONDUCTS NUMERICAL DIFFERENTIATION USING
*     THE 4 POINT CENTRAL DIFFERNCE METHOD.

*     MDATA  = ARRAY CONTAINING POTENTIAL AND LOG I DENSITY
*     SLOPE  = ARRAY CONTAINING DE/D(LOGI) AND LOG I
*     FSLOPE = CENTRAL DIFFERNCE DIFFERENTIATION
*     ECORR1 = UPPERLIMIT OF ANODIC/CATHODIC DISTRIBUTION
*     ECORR2 = LOWERLIMIT OF ANODIC/CATHODIC DISTRIBUTION
*     N      = NUMBER OF DATA POINTS
*     J      = LOCATION WITHIN MDATA OF ECORR
*     I      = CATHODIC BRANCH LOOP COUNTER
*     K      = ANODIC BRANCH LOOP COUNTER
*     H      = DIFFERENTIAL VALUE IN POTENTIAL
*     I1     = VALUE OF LOG I AT E-2H
*     I2     = VALUE OF LOG I AT E-H
*     I3     = VALUE OF LOG I AT E
*     I4     = VALUE OF LOG I AT E+H
*     I5     = VALUE OF LOG I AT E+2H
*     DEDI   = SLOPE VALUE OF I3

      SUBROUTINE DATADEL (MDATA,J,YDAT,YDAT1,XDAT,XDAT1,N2,N1)
      REAL*8 MDATA(2000,2),X1,X2,X3,X4,X5,H,FSLOPE,DEDI
      REAL*4, XDAT1(1000),YDAT1(1000), YDAT(1000),XDAT(1000)
      INTEGER*2 J,K,I
      FSLOPE (H,X1,X2,X4,X5)=(X1+(8.D0*X4)-(8.D0*X2)-X5)/(12.D0*H)

C     PERFORMING CALCULCATIONS ON CATHODIC BRANCH

      K = 0
      DO 100 I = J-2,N1+2,-1
         K = K+1
         H  = MDATA(I,1) - MDATA(I-1,1)
         X1 = MDATA(I-2,2)
         X2 = MDATA(I-1,2)
         X3 = MDATA(I,2)
         X4 = MDATA(I+1,2)
         X5 = MDATA(I+2,2)
         DEDI = 1.D0/FSLOPE(H,X1,X2,X4,X5)
         YDAT(K) = DEDI
         XDAT(K) = MDATA(I,2)
100   CONTINUE
      I = 0

C     PERFORMING CALCULATIONS ON THE ANODIC BRANCH

200   DO 300 K = J+2, N2-2
         I = I + 1
         H  = MDATA(K+1,1) - MDATA(K,1)
```

```
            X1 = MDATA(K-2,2)
            X2 = MDATA(K-1,2)
            X3 = MDATA(K,2)
            X4 = MDATA(K+1,2)
            X5 = MDATA(K+2,2)
            DEDI =  1.DO/FSLOPE(H,X1,X2,X4,X5)
            YDAT1(I) = DEDI
            XDAT1(I) = MDATA(K,2)
300      CONTINUE
400      RETURN
         END
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Research Administration, Code 012 Naval Postgraduate School Monterey, California 93943	2
4.	Professor A. J. Perkins, Code 69Ps Naval Postgraduate School Monterey, California 93943-5000	10
5.	LT. Thomas E. O'Loughlin 11 Ferndale Lane Homer, New York 13077	3
6.	Mr. Robert Hardy, Code 2803 David W. Taylor Naval Ship R&D Center Annapolis, Maryland 21402	3
7.	Ms. Cathy Wong, Code 2812 David W. Taylor Naval Ship R&D Center Annapolis, Maryland 21402	5
8.	Mr. A.G.S. Morton, Code 2813 David W. Taylor Naval Ship R&D Center Annapolis, Maryland 21402	2
9.	Dean of Science and Engineering Naval Postgraduate School Monterey, California 93943	1
10.	Department Chairman, Code 69Hy Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1

No. Copies

11. Professor Y. S. Shin, Code 69Sg — 1
    Department of Mechanical Engineering
    Naval Postgraduate School
    Monterey, California 93943-5000

12. Mr. Charles Zanis, Code 196 — 1
    Naval Sea Systems Command
    Washington, D.C. 20362

13. Mr. V. J. Castelli, Code 2844 — 1
    David W. Taylor Naval Ship R&D Center
    Annapolis, Maryland 21402

IIOOL
J0043-6002

Thesis
04735
c.1

O'Loughlin
Determination of Tafel constants in nonlinear polarization curves.

Thesis
04735
c.1

O'Loughlin
Determination of Tafel constants in nonlinear polarization curves.

AMERICAN BINDERY
TOPEKA KANSAS

thesO4735
Determination of Tafel constants in nonl
3 2768 000 78526 5
DUDLEY KNOX LIBRARY c.1